VIBRATION ANALYSIS OF 3-D PIPING VIA TRANSFER MATRICES

Chester Davis Rudolf

United States
Naval Postgraduate School

THESIS

VIBRATION ANALYSIS OF 3-D PIPING
VIA TRANSFER MATRICES

by

Chester Davis Rudolf, III

December 1971

Vibration Analysis of 3-D Piping via Transfer Matrices

by

Chester Davis Rudolf, III
Lieutenant, United States Navy
B.S., University of Wisconsin, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

This thesis develops the theory for and presents a digital computer program capable of determining the natural frequencies of a three-dimensional piping system having arbitrary configuration. The analysis uses the method of transfer matrices. Piping hangers, loops, and complex branches (branches emanating from branches) have not been included in the analysis. A distributed mass model is used for straight pipe sections, while mass is lumped for curved sections. Inclusion of shear deflection and rotary inertia is optional.

Several piping configurations are analyzed using the program; the results are compared with analytical solutions or values from the literature to demonstrate the accuracy and integrity of the program.

TABLE OF CONTENTS

LIST OF TABLES

I. INTRODUCTION

A. THE PROBLEM

From an engineering standpoint it is important to be able to analyze and predict the vibration characteristics of structural elements and assemblies. The mathematical aspects of several approaches to such analysis have been known for a long time, but the volume of arithmetic operations dictated a gross simplification of complicated systems in order to reduce the arithmetic to manageable limits. The development of the electronic digital computer within the last two decades, however, has provided a means for the practical and accurate analysis of quite complicated vibrating systems without great simplifications which cast doubt on the integrity of results. In turn, this has provided a stimulus for further developments in theory which take advantage of computer capabilities.

Most methods of analysis of actual systems in which both mass and elastic compliance are continuously distributed, involve a discretization of one sort or another; usually the total mass of the system is considered to be lumped at a finite number of distinct points, the masses being connected by massless spring elements. It is almost universally assumed that the elastic characteristics of the system are linear and that the excursions of points in the system from their equilibrium positions are sufficiently small that no

geometric nonlinearities are involved. Such an idealization leads to n second order ordinary differential equations. Provided vibrations are assumed to occur isochronously and no damping is present, these equations reduce to a system of n algebraic equations in the n amplitudes and frequency squared.

The most convenient and expedient way to cast this problem is in matrix form. Thus engineers using these methods deal with such quantities as the mass-matrix, the stiffness matrix, the modal matrix, etc.

On the other hand, for certain simple cases of distributed mass and elasticity, it is not necessary to lump the mass into a finite number of concentrated masses. Instead the vibration problem can be formulated as a partial differential equation which can be solved directly. Recently, this point of view has been extended for the analysis of more complicated cases than can be treated by the conventional use of partial differential equations. Namely, the system being analyzed is divided into a number of sub-systems, a satisfactory analysis existing for each one; then, by a procedure which will be described in greater detail later, these sub-solutions are combined to obtain the solution applicable to the original, complicated configuration. This procedure, known as the transfer matrix method, evolved from a method described by H. Holzer fifty years ago for the analysis of torsional vibrations. Applications of Holzer's viewpoint to the analysis

of flexural systems was made independently by M. A. Prohl and N. O. Myklestad about thirty years ago. The essentials of these methods were abstracted by a number of engineers and mathematicians who developed and formalized them into what is now known as the transfer matrix method. The names of S. Falk, K. Marguerre, E. C. Pestel, and F. A. Leckie, among others, are associated with this development. A book by Pestel and Leckie [Ref. 9] represents the fullest and most accessible treatment of the subject available today.

The transfer matrix method is most naturally applicable to the analysis of "chain-like" structures composed of elements which are adjoined at distinct points and for each of which a satisfactory vibration analysis exists. The simplest application is to the case where such a chain of elements has but two ends. However, it can be applied to structures having slightly greater topological complexity, but it simply is not appropriate for the analysis of structures which are highly reticulated or in which major elements are joined along lines or surfaces rather than at distinct points.

Piping systems are substantially chain-like and topologically simple. Accordingly, it was natural that the transfer matrix method be applied to the analysis of vibration in piping. In this country the first such effort which was reported was that done by G. E. Fink in his Naval Postgraduate School thesis of 1964. At about the same time similar efforts were being made in Japan under the sponsorship of

the Japan Society of Mechanical Engineers. It is known that the Japanese have produced useful digital computer programs for the analysis of piping using the transfer matrix method, but it has not been possible to learn any of the details since these programs remain under the proprietary control of the Japanese government.

Fink developed a program which has convenient generality and gives good results, but which is limited to systems lying entirely in a single plane. His program is capable of making separate determinations of "in-plane" and "out-of-plane" vibrations, which, for the case of uniplanar configurations, are uncoupled. Further work on the theory was done by W. S. Baird, Jr. and J. L. Simmons in their Naval Postgraduate School theses; this work mainly related to systems having appreciable topological complexity. In his Naval Postgraduate School thesis, Y. S. Kim returned to the development of Fink's program, adding several useful features, including the incorporation of several alternative mathematical techniques for accurate solution. Some of these techniques reduced solution time below that required in Fink's original program; however, Kim's program is still limited to uniplanar systems.

Accordingly, the writer undertook to develop a program capable of treating truly three-dimensional configurations. The remainder of this thesis presents the analytical background of this development, lists the program which includes

rules for its operation, and gives evidence concerning the high accuracy of the results.

It should be made clear that no claims are made concerning the relative usefulness of the transfer matrix method as compared to lumped-mass methods which are presently in widespread use for analysis of piping vibrations. In general, the writer believes that the results which are obtained by the transfer matrix procedure are substantially more accurate than can be obtained in most cases using lumped-mass analysis. However, the delicacy of the calculations involved in the transfer matrix method seems to imply that only a limited number of the lowest frequencies (and their associated modes) can be obtained using present programming methods and computer hardware. Greater word-length in the computer, or availability of fast multiple precision capability, would permit obtaining a larger number of frequencies and modes via the transfer matrix method. Briefly, the motivation for the development reported herein has been simply that the transfer matrix method is clearly appropriate for the analysis of piping vibrations and that the application should be made. It seems likely that in some cases and for some purposes the transfer matrix method may be more convenient and economical while in other cases lumped-mass methods may be preferable.

B. THE TRANSFER MATRIX METHOD

Specific details of the application of transfer matrix methodology to the analysis of piping vibrations will be given in subsequent parts of this thesis. Background material and a general exposition of the method is most readily available in the book by Pestel and Leckie [Ref. 9] and it would be pointless to include herein a paraphrase of such material. However, for the reader who is not familiar with the ideas behind the transfer matrix method, a very brief exposition is given in this section.

An appropriate "state vector" is defined which presents force-type and deflection-type information capable of specifyint the "state" at each of several points in the configuration at which sub-elements are joined together. These quantities specify the configuration and internal force system in the structure at its extreme isochronous, deflected position. For example, in a simple torsional system, there are only two quantities in the state vector, the angle of rotation from the equilibrium position and the torque transmitted from one element to the next. The elastic compliance and the inertial characteristics of an element permit obtaining equations relating the quantities in the state vector at the left end of the element to those in the state vector at the right end of the element. When these relations are put in matrix form it is found that the state vector at the right can be written as the product of the state vector at the left premultiplied by a square

matrix, called a transfer matrix, the elements of which represent the elastic and inertial properties of the element.

In applying the method, a state vector is chosen for the left end of the assembly of elements in such a way as to satisfy conditions of restraint at the left. Then, by successive multiplication by transfer matrices representing individual elements encountered in order, progressing from the left end of the assembly toward the right, one arrives at a representation for the state vector at the right end. The individual transfer matrices are functions of isochronous frequency; accordingly the right end matrix is a function of frequency. When frequency is adjusted, any particular frequency which permits satisfaction of restraint at the right end is a natural frequency of the system. The several state vectors at the junction points give the deflected configuration, or mode shape, corresponding to this frequency, when this value is substituted in the expressions for the state vectors.

For simple cases it is possible to carry out the multiplications in literal form, keeping the frequency as an unknown parameter. Then the right end restraint conditions provide a polynomial equation in the frequency which can be solved by customary methods. However, in relatively complicated cases it is not feasible to retain the frequency as a literal parameter. In these cases a definite numerical value is assumed for frequency and this leads to a numerical measure of the failure to satisfy right end restraint

conditions. By assuming different values of the frequency and by constructing a curve of measure-of-failure versus frequency, the correct frequencies can be obtained.

The transfer matrices which describe the individual elements may be a theoretically complete and full representation, or they may themselves by approximate representations. In the application to piping systems it is convenient to represent a straight length of pipe by a transfer matrix which is theoretically complete and exact. For systems composed of only straight lengths of pipe there is no approximation whatsoever in the transfer matrix method (other than that implied by internal round-off in the computer); this may be compared with the fact that all lumped-mass procedures definitely imply an approximation and thus result in some theoretical error.

Curved pipe elements (elbows and bends) could be treated exactly since adequate theory exists for the construction of theoretically exact transfer matrices with which to represent them. However, there are severe computational difficulties involved with such exact representation. Accordingly, with the awareness that the total length of curved pipe is but a small fraction of the total length of both straight and curved pipe in the average configuration, it was decided to represent curved pipe by a succession of massless curved sections having proper elastic properties, alternating with appropriately located point masses. For each of these idealized elements, it is

possible to obtain the corresponding transfer matrices with no computational difficulties. Thus, elbows and bends are represented in much the same way that is done with the lumped-mass methods mentioned previously. It is believed that the approximation introduced by this point-mass representation of a small part of a configuration is quite tolerable for practical engineering purposes. However, it is certainly possible to modify the program reported herein so as to provide for exact representation for curved elements. This would not involve major changes but only the development of a subroutine to replace those given herein for curved elements.

II. PROGRAM DEVELOPMENT

A. THE STATE VECTOR, TRANSFER MATRIX, AND FREQUENCY DETERMINANT

1. State Vector

The state vector z_i at point i of an elastic system is a column vector whose components are the generalized displacements, D_i, and the generalized forces, F_i, at that point. In matrix notation, we have

$$z_i = \begin{Bmatrix} D \\ F \end{Bmatrix}_i$$

For a beam free to move in three dimensions with longitudinal, torsional and flexural elasticity, the state vector takes the form

$$z_i = \begin{cases} u & \text{- deflection in x-direction} \\ v & \text{- deflection in y-direction} \\ w & \text{- deflection in z-direction} \\ \phi & \text{- rotation about x - axis} \\ \psi & \text{- rotation about y-axis} \\ \theta & \text{- rotation about z-axis} \\ N & \text{- axial force in x-direction} \\ V_y & \text{- shear force in y-direction} \\ V_z & \text{- shear force in z-direction} \\ T & \text{- Torque about x-axis} \\ M_y & \text{- Moment about y-axis} \\ M_z & \text{- Moment about z-axis} \end{cases}$$

The order in which the individual displacements and forces occur in the state vector is of no particular importance so long as consistency is maintained. Program VIBREL

(for VIBRation of ELastic piping), the program developed herein, employs the arrangement given above.

2. Sign Convention and Coordinate System

A right-handed cartesian coordinate system is used to describe the local generalized forces and displacements, the x-axis coinciding with the centroidal axis of the pipe as shown in Fig. 2.1.

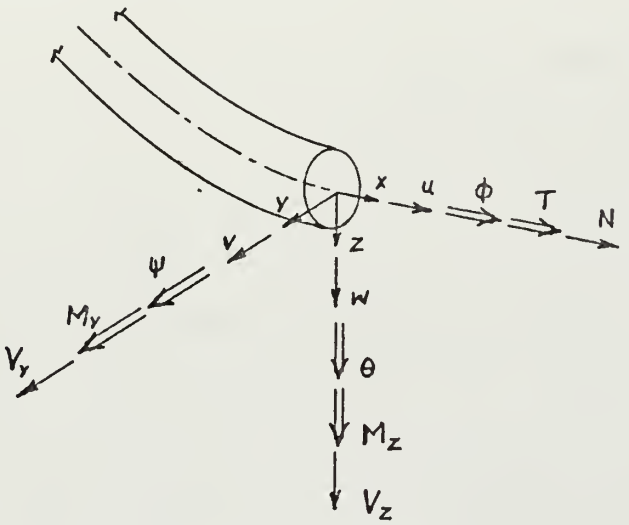


Figure 2.1: Vectorial Representation of State Vector Displacement and Forces

A cut made at any location of the pipe will expose two faces; the face with its outward normal in the positive x-axis direction is considered positive. The state vector parameters are positive if their vector representations coincide with the directions prescribed by the cartesian coordinate system at the positive face as in Fig. 2-1. Rotation and moment vectors are portrayed according to the right-hand-screw rule by a double line with an arrowhead.

3. Transfer Matrices

The transfer matrix is defined as the matrix which relates two state vectors at successive selected points of

division in the elastic system.

$$z_i = U_i z_{i-1}$$

where U_i is the transfer matrix. The transfer matrix must be square, and in the case of a spatial system must have 12 rows and 12 columns. When the transfer matrix relates the flexibility between the points i and i-1 of a continuous system it will hereafter be referred to as a field matrix, V. When a discontinuity arises in either force or displacement, as with a point mass in a lumped system, the transfer matrix relating the state vectors on either side of the discontinuity is called a point matrix, P. The term "transfer matrix" will henceforth be used with regard to the matrix, U, which includes mass and flexibility in the relationship between the state vectors at two distinct points. A complete discussion of transfer matrices, with examples and derivations, can be found in Ref. 9.

4. The Elimination Process and Frequency Determinant

The process of combining the system transfer matrices to find natural frequencies can be demonstrated by a simple planar example using the Myklestad method of replacing a distributed mass beam with lumped masses connected by flexible elements (Fig. 2.2). Assuming that the field and point

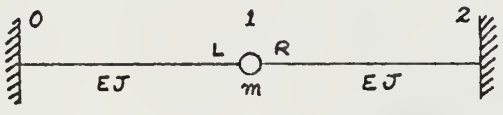


Figure 2.2: Lumped Mass Representation of a Planar Beam

matrices for a massless flexible beam and a point mass are already known, then

$$z_1^L = V_1 z_o, \quad z_1^R = P z_1^L, \quad \text{and} \quad z_2 = V_2 z_1^R$$

where $z_i = \begin{Bmatrix} w \\ \psi \\ V_z \\ M_y \end{Bmatrix}_i$ and superscripts L and R indicate left side or right side, respectively. Combining the above yields the relationship

$$z_2 = V_2 P V_1 z_o = U z_o \tag{2-1}$$

where U is the system transfer matrix. Written in full this relationship is

$$\begin{Bmatrix} w \\ \psi \\ V_z \\ M_z \end{Bmatrix}_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} w \\ \psi \\ V_z \\ M_z \end{Bmatrix}_0$$

the a_{ij} being vibration functions of mass, stiffness, and frequency. Application of the boundary conditions $w_o = w_2 = \psi_o = \psi_2 = 0$ reduces the matrix equation to the following

$$\begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix} \begin{Bmatrix} V_z \\ M_z \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} . \tag{2-2}$$

Because the elements of this square matrix are functions of frequency, its determinant is called the frequency determinant of the system. Note that this 2 X 2 matrix is a

submatrix of the 4 X 4 matrix appearing above. A non-trivial solution of the equations 2-2 requires that the value of the frequency determinant (Δ) equal zero. A value of the frequency, ω, which causes Δ to vanish is called a natural frequency of the system.

The natural frequencies could be solved for directly provided the expressions for the a_{ij} were kept in terms of the circular frequency, ω, but in practice these expressions are forms much too unwieldy to be handled explicitly. Program VIBREL evaluates the frequency determinant of order 6 X 6 for successive numerical values of frequency, zero values of the frequency determinant corresponding to natural frequencies of the system.

Let the state vector be composed of 2r quantities. Then to compute U in Eq. 2-1, two multiplications of 2r X 2r matrices are required. Noticing that application of the boundary conditions to the state vector at point 0 will cause r of its elements to be zero, we find that r columns of U are multiplied by zero, and thus play no part in the calculation. Kim [Ref. 6] introduced the concept of the "state matrix," S, such that the left end state vector, z_o, of 2r quantities can be written in terms of a compressed state vector, z_o^*, having only r quantities, viz.,

$$z_o = Sz_o^*$$

where S is the state matrix at the left end, having r columns and 2r rows, its elements consisting only of zeros

and ones. All elements of the state matrix are zero except that in each column, unity appears in the same row position as the corresponding non-zero term in the state vector, boundary conditions having been applied. If the state matrix is used in the multiplication scheme, then we can multiply 2r X r matrices by 2r X 2r matrices and reduce the calculation considerably. The resulting 2r X r system transfer matrix requires only the application of right end boundary conditions to yield the frequency condition.

To illustrate this state matrix concept most simply, consider again a two-dimensional case in which the state vector consists of four quantities. The three dimensional case is an obvious generalization of the following discussion. Taking strictly the state vector approach we can

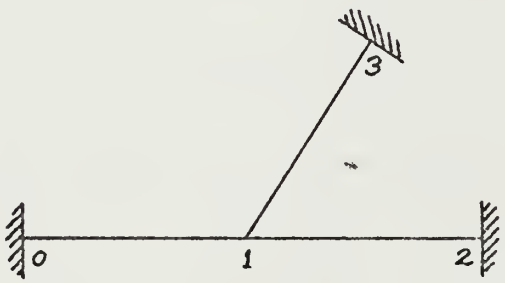


Figure 2.3: Single Branch Planar System

express the relation between the state vector at 2 and that at 0 by

$$z_2 = U_2 P_1 U_1 z_o \qquad (2\text{-}3)$$

where U_2 and U_1 are straight section transfer matrices and the point matrix P_1 accounts for the effects of the branch. Carrying out the matrix multiplication and exhibiting the

system transfer matrix and state vectors which consider bending only, we have

$$\begin{Bmatrix} w \\ \psi \\ V_z \\ M_z \end{Bmatrix}_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ A_1 \\ A_2 \end{Bmatrix} \tag{2-4}$$

The boundary conditions have been applied to the left end state vector; the quantities A_1 and A_2 represent the unknown quantities of V_z and M_z respectively. Application of the right end boundary conditions yields a frequency condition of the same form as Eq. 2-2.

Using the state matrix approach instead, Eq. 2-3 becomes

$$z_2 = U_2 P_1 U_1 S z_o^* \tag{2-5}$$

where S is the state matrix $\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. z_o^* contains the non-zero quantities of the left end state vector. After multiplication of the matrices of Eq. 2-5, the number of individual multiplications being considerably less than before, the following relation is obtained:

$$\begin{Bmatrix} w \\ \psi \\ Y_z \\ M_z \end{Bmatrix}_2 = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \end{Bmatrix} \tag{2-6}$$

We see from this result that inclusion of the state matrix compresses the left end state vector to r quantities while providing the same useful information as the matrix of Eq. 2-4, inasmuch as the same frequency condition is obtained after the right end boundary conditions are applied. Also, if the relative magnitudes of the unknowns A_1 and A_2 have been determined, the mode deflections and forces at a point can be calculated simply by multiplication of the compressed left end state vector, consisting of r quantities, by the 2r X r system transfer matrix up to that point. The appropriate mode frequency is used in the computation. This will be discussed in greater detail in Chapter 3.

B. 3-D PIPING GEOMETRY

1. Working Points and Piping Nomenclature

The geometry of the piping system as it is used in program VIBREL is based on working points. A working point of the system occurs where cross section or properties change, where branches join the main member, where the piping changes direction, and at the ends of the main member and branches. These points are measured in cartesian coordinates from an origin located arbitrarily but fixed in space. For purposes of this discussion we shall distinguish between the two types of working points at which piping direction changes. At a corner point (Fig. 2.4) the piping direction changes with no discernible radius of curvature. At a bend point (Fig. 2.5), which is defined as the intersection

of the tangents to the two ends of a curved pipe, the radius of curvature is finite.

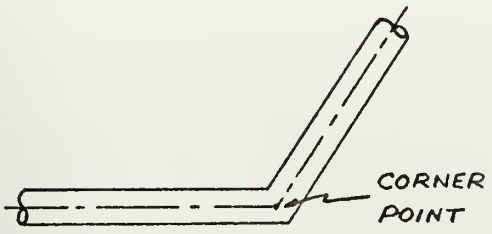


Figure 2.4: Piping Corner

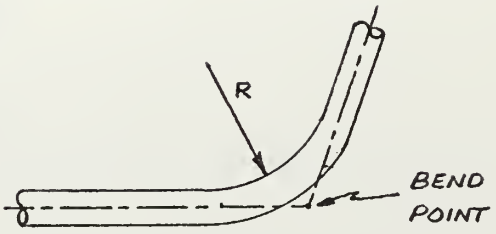


Figure 2.5: Piping Bend

The main member is designated as the longest continuous run of pipe from which all branches emanate. Program VIBREL cannot handle branches emanating from branches.

2. Unit Vectors and Coordinate Transformations

If a set of mutually orthogonal x, y, and z unit vectors are formed at the main member and branch starting points, and prior to and following every every directional change, we can determine, by angular differences in successive sets of unit vectors, the appropriate coordinate transformation angles. The local coordinate systems described by these unit vectors are formed with the piping system in its quiescent configuration.

Let us take, for example, the two sections of pipe pictured in Fig. 2.6 determined by the three working points E, F, G, where F is a corner point. Also let the two sections of pipe, EF and FG, be represented by vectors $\vec{A}$ and $\vec{B}$, the

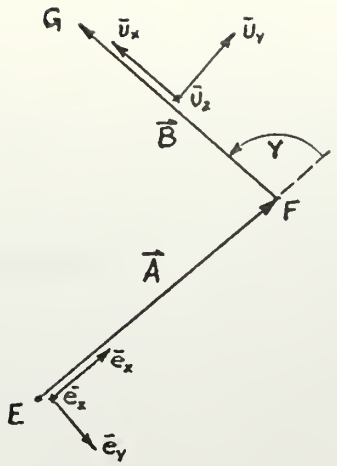


Figure 2.6: Vector Representation of Directional Change

components of which have been determined from the coordinates of E, F, and G. Then the angle γ can be determined by the following relation:

$$|\gamma| = \text{Cos}^{-1}\left[\frac{\bar{B} \cdot \bar{A}}{|\bar{B}|\ |\bar{A}|}\right]$$

The sign will be determined later. Since the vectorial representation of state vector quantities in Fig. 2.1 requires that the x-axis coincide with the centroidal axis of the pipe, let

$$\bar{e}_x = \frac{\bar{A}}{|A|}$$

For the purposes of succeding derivations we desire that $\bar{e}_y$ be in the plane of $\bar{A}$ and $\bar{B}$ and point outward from the directional change. Hence

$$\bar{e}_z = \frac{\bar{B} \times \bar{A}}{|\bar{B} \times \bar{A}|} \quad \text{and} \quad \bar{e}_y = \bar{e}_z \times \bar{e}_x$$

By the right hand screw convention the rotation of amount γ will always be in the negative $\bar{e}_z$ sense. The set of unit

vectors, $\bar{U}_x$, $\bar{U}_y$, $\bar{U}_z$ are found simply by rotating the initial unit vectors about the local z-axis by the angle γ.

Now consider the difference in orientation between the $\bar{U}_x$, $\bar{U}_y$, $\bar{U}_z$ vectors and those prior to the next directional change farther down the pipe, $\bar{U}_{x'}$, $\bar{U}_{y'}$, and $\bar{U}_{z'}$. These have been constructed in a manner similar to $\bar{e}_x$, $\bar{e}_y$, and $\bar{e}_z$. Referring to Fig. 2.7, the angular difference between the

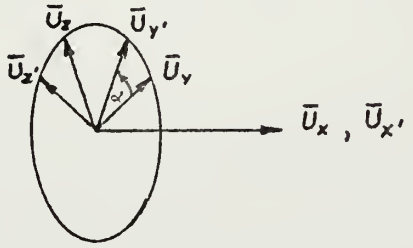


Figure 2.7: Unit Vector Rotation about x-axis

two adjacent sets of unit vectors is merely the rotation α about the local x-axis. Alpha may be determined by the expression

$$|\alpha| = \text{Cos}^{-1} \ (\bar{U}_y \cdot \bar{U}_{y'})$$

This rotation is in the positive (negative) $\bar{U}_x$ sense according to whether the triple product $(\bar{U}_y \times \bar{U}_{y'}) \cdot \bar{U}_{x'}$ is positive (negative).

In this way, the state vector is always expressed in the appropriate local coordinate system. For the simple system of Fig. 2.8

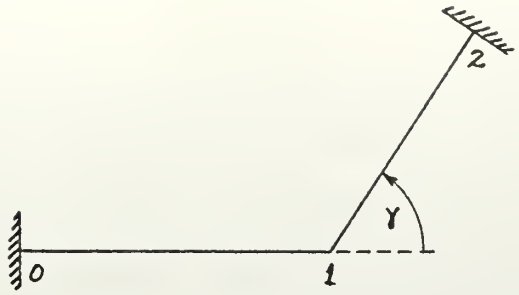


Figure 2.8: Planar System with Directional Change

$$z_1^L = U_1 z_o \quad z_1^R = U_\gamma z_1^L \quad z_2 = U_2 z_1^R$$

which combining gives

$$z_2 = U_2 U_\gamma U_1 z_o$$

where U_γ is the transformation matrix corresponding to a rotation γ about the z-axis. For a 3-D system with the state vector defined as in section 2.1 this matrix is

$$U_\gamma = \text{DIAG}\,[T_\gamma, T_\gamma, T_\gamma, T_\gamma] \text{ , where } T_\gamma = \begin{bmatrix} \text{COS}\,\gamma & \text{SIN}\,\gamma & 0 \\ -\text{SIN}\,\gamma & \text{COS}\,\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly, for the x-axis rotations

$$U_\alpha = \text{DIAG}\,[T_\alpha, T_\alpha, T_\alpha, T_\alpha] \text{ , where } T_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{COS}\,\alpha & \text{SIN}\,\alpha \\ 0 & -\text{SIN}\,\alpha & \text{COS}\,\alpha \end{bmatrix}$$

C. 3-D TRANSFER MATRICES

1. The Straight Pipe Transfer Matrix

In order to maximize accuracy, a distributed mass model was decided upon for straight pieces of pipe. The program uses the straight bar transfer matrix cited in the appendix of Ref. 9, but revised to conform to the ordering

of the state vector described herein. The revised matrix, which appears in Appendix B, combines the vibration functions for longitudinal, torsional, and flexural modes and can include or neglect rotary inertia and shear deflection.

2. Development of the Curved Pipe Field Matrix

Due to the complexity and increased machine time involved in consideration of a distributed mass model for curved sections of pipe, a lumped mass treatment was chosen. This, coupled with the distributed mass approach for straight pipe, yields staisfactory accuracy combined with minimal computer time.

Given the compliance of a piece of massless curved pipe at its center of curvature, we seek the field matrix V such that $z_R = Vz_L$ (see Fig. 2.9) where $z_i = \{\begin{matrix} D_i \\ F_i \end{matrix}\}$. The

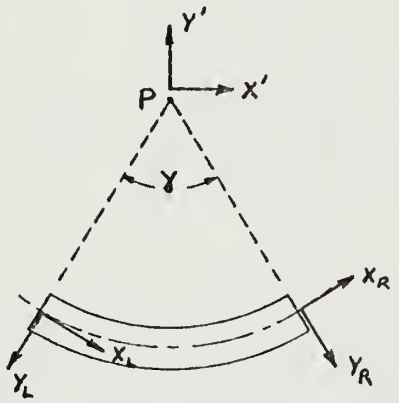


Figure 2.9: Massless Curved Pipe

compliance matrix C_p is defined by the relationship $D_p = C_p F_p$ where D_p is a vector of generalized displacements observed at P and F_p is some vector of generalized forces applied at P. Point P is assumed to be attached by a rigid

bar to one end of the pipe while the other end of the pipe remains fixed. The compliance matrix for a pipe bend and its derivation are presented in a paper by J. E. Brock [Ref. 2].

The analysis requires that we find the compliance matrix at point L which is obtained by the appropriate rotation and translation of coordinates from point P such that

$$C_L = B_{PL}^T \, L_{PL}^T \, C_P \, L_{PL} \, B_{PL}$$

where $B_{PL} = \begin{bmatrix} I & 0 \\ b_{PL} & I \end{bmatrix}$, $b_{PL} = \begin{bmatrix} 0 & z_P - z_L & y_L - y_P \\ z_L - z_P & 0 & x_P - x_L \\ y_P - y_L & x_L - x_P & 0 \end{bmatrix}$

for translation and

$$L_{PL} = \begin{bmatrix} K_{PL} & 0 \\ 0 & K_{PL} \end{bmatrix} , \quad K_{PL} = \begin{bmatrix} \cos(x_P,x_L) & \cos(x_P,y_L) & \cos(x_P,z_L) \\ \cos(y_P,x_L) & \cos(y_P,y_L) & \cos(y_P,z_L) \\ \cos(z_P,x_L) & \cos(z_P,y_L) & \cos(z_P,z_L) \end{bmatrix}$$

for rotation.

$$L_{PL} = L_{LP}^T \quad \text{and} \quad B_{PL} = B_{LP}^{-1}$$

The matrix equation for the generalized displacements at point L can now be written.

$$D_L = C_L F_L + B_{RL}^T \, L_{LR} \, D_R$$

where the rotation and translation matrices use the above notation. Solving for D_R we arrive at

$$D_R = L_{LR}^T B_{LR}^T D_L - L_{LR}^T B_{LR}^T C_L F_L \qquad (2\text{-}7)$$

The equation for the generalized forces at the right end of the pipe has the form

$$F_R = -L_{RL} B_{RL} F_L \quad . \qquad (2\text{-}8)$$

If equations 2-7 and 2-8 are combined in matrix form, we have the result

$$\begin{Bmatrix} D_R \\ F_R \end{Bmatrix} = \left[\begin{array}{c|c} L_{LR}^T B_{LR}^T & -L_{LR}^T B_{LR}^T C_L \\ \hline 0 & -L_{RL} B_{RL} \end{array}\right] \begin{Bmatrix} D_L \\ F_L \end{Bmatrix}$$

By the sign convention discussed in section 2.1, point L is located at a negative face, so after appropriate sign changes are taken into account, the field matrix for the bend becomes

$$V = \left[\begin{array}{c|c} L_{LR}^T B_{LR}^T & L_{LR}^T B_{LR}^T C_L \\ \hline 0 & -L_{RL} B_{RL} \end{array}\right]$$

This matrix in its explicit form, as used in program VIBREL, appears in Appendix B.

3. Development of Curved Pipe Point Matrix

The mass of a pipe bend or elbow is taken into account through the transfer matrix procedure by lumping it at the center of mass and including its effect on the system at point O (Fig. 2.10), the left end of the elbow section. The resulting point matrix relates only the forces across point O since the deflections are continuous.

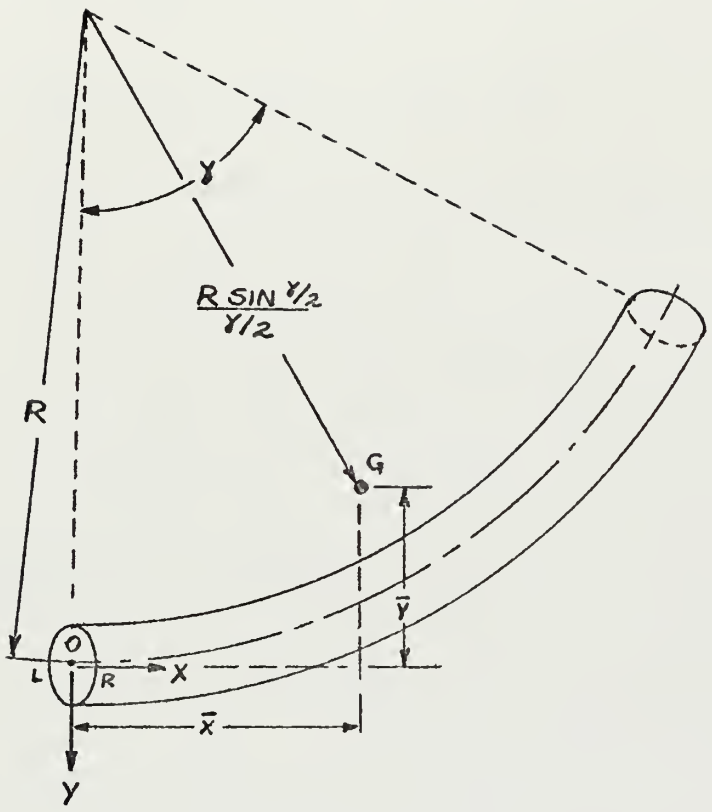


Figure 2-10: Curved Pipe Showing Location of Center of Mass

A force summation at point O yields the following scalar equations (the quantities appearing in this development are defined in Appendix A, subscripts L and R refer to the left and right of point O, respectively),

$$N_R = N_L - m\omega^2(u - \bar{y}\theta) \tag{2-9}$$

$$V_{yR} = V_{yL} - m\omega^2(v + \bar{x}\theta) \tag{2-10}$$

$$V_{zR} = V_{zL} - m\omega^2(w - \bar{y}\phi - \bar{x}\psi) \tag{2-11}$$

Similarly, a moment summation at the same point produces

$$T_R = T_L - I_{xo}\,\omega^2\phi \tag{2-12}$$

$$M_{yR} = M_{yL} - I_{yo}\,\omega^2\psi \tag{2-13}$$

$$M_{zR} = M_{zL} - I_{zo}\,\omega^2\theta \tag{2-14}$$

Combining equations 2-9 through 2-14

$$F_R = TF_L$$

where

$$T = \begin{bmatrix} -m\omega^2 & 0 & 0 & 0 & 0 & -m\omega^2\bar{y} \\ 0 & -m\omega^2 & 0 & 0 & 0 & -m\omega^2\bar{y} \\ 0 & 0 & -m\omega^2 & m\omega^2\bar{y} & m\omega^2\bar{x} & 0 \\ 0 & 0 & 0 & -I_{xo}\omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -I_{yo}\omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I_{zo}\omega^2 \end{bmatrix}$$

and the resulting point matrix equation is:

$$\begin{Bmatrix} D_R \\ F_R \end{Bmatrix} = \begin{bmatrix} I_6 & O_6 \\ T & I_6 \end{bmatrix} \begin{Bmatrix} D_L \\ F_L \end{Bmatrix}$$

An inertia tensor analysis was used in the determination of the mass moments of inertia about point O. It can readily be shown that the inertia tensor at point O is

$$\bar{\bar{I}}_o = \frac{\rho A \bar{r}^2}{2}[\gamma \bar{\bar{u}} + (\frac{\gamma}{2} + \frac{SIN2\gamma}{4})\bar{i}\bar{i} - SIN^2\gamma(\frac{\bar{i}\bar{j}+\bar{j}\bar{i}}{2}) + (\frac{\gamma}{2} - \frac{SIN2\gamma}{4})\bar{j}\bar{j}]$$

$$+2\rho AR^3[\gamma - SIN\gamma]\bar{\bar{u}} + \rho AR^3[(-\frac{\gamma}{2} + SIN\frac{2\gamma}{4})\bar{i}\bar{i}$$

$$+(1-COS\gamma - \frac{SIN^2\gamma}{2})(\bar{i}\bar{j}+\bar{j}\bar{i}) - (\frac{3\gamma}{2} - 2SIN\gamma + \frac{SIN2\gamma}{4})\bar{j}\bar{j}]$$

where it is assumed that pipe wall thickness is small compared to pipe radius. (The symbols used here are defined in Appendix A.) Then $I_{xo} = \vec{i} \cdot \bar{\bar{I}}_o \cdot \vec{i}$, $I_{yo} = \vec{j} \cdot \bar{\bar{I}}_o \cdot \vec{j}$, and $I_{zo} = \bar{k} \cdot \bar{\bar{I}}_o \cdot \bar{k}$, where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors along the local coordinate axes at point O. The ensuing moments of inertia and point matrix are presented in their entirety in Appendix B.

D. BRANCHED SYSTEMS

Any branch joining a piping system has a significant effect on its dynamic behavior. While the deflections across the junction point are continuous, there is a discontinuity in the forces, the magnitude of which is dependent on the displacement at the junction point and the nature of the branch.

Consider the simple branch system of Fig. 2.11.

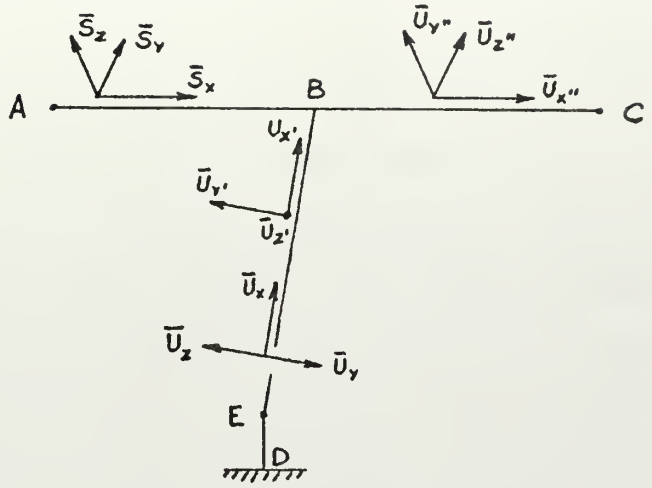


Figure 2.11: Unit Vector Orientation at a Branch Junction

Using the state matrix concept discussed in section 2.1 we have the following relationship between the state vector at point D and the state vector at point B (the bar denotes branch coordinate scheme).

$$\left\{\begin{matrix}\bar{D}_B\\ \bar{F}_B\end{matrix}\right\}^{12\ X\ 1} = \left[\begin{matrix}R_1^{6\ X\ 6}\\ R_2\end{matrix}\right]\{\bar{z}_D\}^{6\ X\ 1} \qquad (2\text{-}15)$$

The zero quantities of the state vector at D have been eliminated by application of the boundary conditions and introduction of the state matrix, and the branch transfer matrix has been partitioned into two submatrices R_1 and R_2. The local coordinate system of the branch at point B is represented by the unprimed set of unit vectors. From the relationship 2-15 it can be seen that $\bar{D}_B = R_1\bar{z}_D$ and $\bar{F}_B = R_2\bar{z}_D$, so that when $\bar{z}_D$ is eliminated, the equation between the forces and displacements at point B in the unprimed coordinate system becomes $\bar{F}_B = R_2R_1^{-1}\bar{D}_B$ (2-16).

The vectors of the generalized forces and displacements in the branch system must now be transferred to the main member system represented by the unit vectors $\bar{s}$ in Fig. 2.11. This can be achieved by one z-axis and two x-axis transformations of the unprimed coordinates. We first form the primed set of unit vectors in a manner identical to that for a corner point (section 2.2), the three working points describing the corner being E, B and C. An x-axis rotation α_1 will align the unprimed with the primed coordinates. Next, a rotation γ about the z'-axis will allow $u_{x'}$ to coincide with the centoidal axis of section BC resulting in the double-primed coordinate scheme. Finally, rotation through the angle α about the x''-axis will align the branch system with the main member coordinates, s. The total transformation of the displacements can be expressed by

$$D_B = G_\alpha \, G_\gamma \, G_{\alpha_1} \, \bar{D}_B = G\bar{D}_B \qquad (2\text{-}17)$$

where $G_\alpha = \text{DIAG}\left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{COS}\alpha & \text{SIN}\alpha \\ 0 & -\text{SIN}\alpha & \text{COS}\alpha \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{COS}\alpha & \text{SIN}\alpha \\ 0 & -\text{SIN}\alpha & \text{COS}\alpha \end{bmatrix} \right\}$,

similarly for G_{α_1}, and

$$G_\gamma = \text{DIAG}\left\{ \begin{bmatrix} \text{COS}\alpha & \text{SIN}\alpha & 0 \\ -\text{SIN}\alpha & \text{COS}\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \text{COS}\alpha & \text{SIN}\alpha & 0 \\ -\text{SIN}\alpha & \text{COS}\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\}.$$

Likewise for the forces, $F_B = G\bar{F}_B$ (2-18). Rearranging relations 2-17 and 2-18 and substituting in 2-16 yields

$$G^{-1}F_B = R_2R_1^{-1}G^{-1}D_B \quad .$$

Noting that $G^{-1} = G^T$ we now form the force displacement equation

$$F_B = GR_2R_1^{-1}G_{\alpha 1}^T G_\gamma^T G_\alpha^T D_B \quad .$$

Referring to the free body diagram in Fig. 2.12 we find

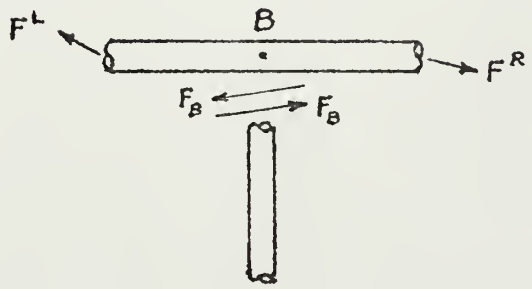


Figure 2.12: Generalized Forces at a Branch Junction

that the main member force equation at point B can be written

$$\begin{aligned} F^R &= F^L + F_B \\ &= F^L + GR_2R_1^{-1}G_{\alpha 1}^T G_\gamma^T G_\alpha^T D_B \quad . \end{aligned}$$

Thus, since $D_B = D^L = D^R$, the point matrix relating the state vectors to the left and right of a branch junction point of a 3-D system is given by

$$\begin{Bmatrix} D \\ F \end{Bmatrix}^R = \left[\begin{array}{c|c} I & 0 \\ \hline GR_2R_1^{-1}G_{\alpha 1}^T G_\alpha^T G_\alpha^T & I \end{array}\right] \begin{Bmatrix} D \\ F \end{Bmatrix}^L$$

The foregoing development assumes that there is no curvature of the main member or branch at point B and that no abrupt change of direction of the main member occurs at that immediate location.

For two or more branches joining the main member at a single point, the system transfer matrix just prior to the branch point is simply multiplied in turn by the point matrix for each of the branches.

E. COMPUTER IMPLEMENTATION OF TRANSFER MATRICES

It would be an extremely tedious job to attempt explicit assemblage of the elements of the system transfer matrix in terms of the circular frequency. Subsequent solution for the roots of the expanded frequency determinant would also be impractical for all but the simplest of systems. It is for these reasons that digital computation becomes a necessity.

Program VIBREL, which incorporates the theory of preceding sections, was developed to utilize the speed with which the complete system transfer matrix can be formed time after time for various values of frequency. A typical graph of frequency determinant $\Delta(\omega)$ versus circular frequency ω is shown in Fig. 2.13. The values of ω at which the graph

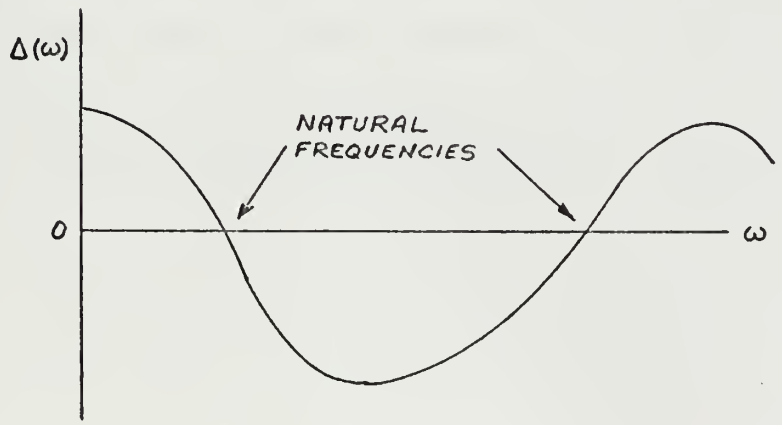


Figure 2.13: Graph of Frequency Determinant vs. Frequency

crosses the horizontal axis, namely those values of ω for which $\Delta(\omega) = 0$, are the natural frequencies of the system.

The program, presented in Appendices C through E, was coded using Fortran language [Ref. 7] in double precision to minimize the effects of round-off and truncation errors. Since the prime objective was to produce a working program of demonstrable accuracy and reliability, limited attention was focused on the niceties of programming intended to reduce execution time. However, Kim's state matrix concept is included and this effects a substantial saving of time as compared to other procedures. There is no reason to believe that the actual programming is particularly inefficient; however, it is not unlikely that close scrutiny of program details might point out some places where slight revisions would effect economies.

The transfer matrices used in computation of system frequencies are those presented in Appendix B.

Input formats were designed to provide satisfactory simplicity for the user in preparing data decks, while output formats were chosen to check for correct input as well as to provide a detailed summary of results. The I/O formats are discussed more thoroughly in Appendix E.

III. DISCUSSION

A. SCOPE OF THE SOLUTION

In its present form, program VIBREL is capable of analyzing a piping system having few restrictions regarding topology. The system must, however, be composed of a main member from which can emanate as many as fifty branches. The program as listed in Appendix E can accommodate no more than two branches joining the main member at a particular point. In addition, there can be no branches emanating from other branches, nor can there by any curvature of the main member or branch at a branch junction point.

The total number of sections capable of being analyzed, presently set at one hundred, can be increased by following the procedure described in Appendix C. Likewise the number of branches can be augmented from its present limit of fifty. For each increase of one section, eighty additional bytes of computer storage are needed above the 140,000 bytes now required for the entire program. For most computer installations this allows considerable leeway in the size of the piping system which can be handled by the program.

To provide an idea of the computer time involved in the frequency analysis, the twelve-section piping system of the sample problem in Appendix G required about four minutes for computation of four modes and almost fifteen minutes for eighteen modes. The IBM 360 computer compiles the program in thirty-nine seconds.

The usual assumptions of linearity as well as homogeneity and isotropy of the structural material have been incorporated in the analysis. All sections of the system are assumed to vibrate isochronously with negligible damping.

B. NATURE OF ERRORS AND INACCURACIES

1. Round-Off Errors

As piping system complexity increases the number of multiplications required to form the system transfer matrix grows proportionately. Each time a multiplication occurs, there is some round-off and loss of accuracy; this is a function of the significant figure capacity of the computer.

FORTRAN, in conjunction with the IBM 360 and double precision arithmetic, has the capability of carrying numbers of 16 significant digits with an exponent range of 10^{-78} to 10^{78}. Simpler systems can be checked for accuracy and it is evident that this significant digit capacity keeps round-off errors negligible. Since with more complex systems other forms of errors creep into the results, we have no way of predicting the exact contribution of round-off errors to total inaccuracy of frequency.

It is reasonable to assume that round-off plays a more significant role in the 3-D piping system accuracy than in a planar case with equivalent sections because of the greater number of coordinate transformations that are necessary for system description.

2. Lumped Mass Idealization

For sections of curved pipe, lumped mass connected by massless springs is used as a model for the real system. This idealization leads to inherent inaccuracy in the analysis of systems containing bends.

3. Zeros of the Frequency Determinant

Once it has been established that the frequency determinant has crossed the zero axis, Program VIBREL uses Mueller's method of successive bisection and inverse parabolic interpolation to determine the zero. Care should be exercised in considering as significant only the same number of places as in the acceptability criterion which was specified for the solution. (See Appendix C.)

4. Other Sources of Inaccuracies

The imperfect nature of the real system and inevitable errors in the measurement of working points will render the computer solution only a good engineering approximation of what can be expected in actuality. In the discussion of accuracy in succeeding sections, inaccuracies caused by real system imperfections will not be considered.

C. ACCURACY AND INTEGRITY OF SOLUTION

1. Solution Accuracy

Confidence in the accuracy of the program solution was established by three comparison tests, the results of which were tabulated in Appendix F.

a. Analytical Comparison

For single straight sections of pipe a closed form solution was obtained from the governing fourth order differential equation with the appropriate fixed, free, or ball joint end conditions. The largest variation between the natural frequencies calculated analytically and those computed by program VIBREL was 0.009%.

b. Reference Value Comparison

Where the exact closed form solution was not available, recourse to the literature provided some comparison frequencies. For the simple curved pipe shown in Appendix F, Refs. 5, 6, and 8 contained a range of frequency values for the first two modes. VIBREL results for the same curved pipe fell within this range in each of the two modes examined, with an average variation from the reference values of about 4%. Because, for a curved pipe, program VIBREL includes bend flexibility factor and an accurate value for shear distribution factor taken from Ref 2, the values computed are considered to be at least as accurate as those contained in the literature.

For a single branch system composed of straight pipes only, comparison frequencies for three modes were available from Ref. 6. The largest difference between reference and VIBREL in this case was 0.06%.

c. Dual Analysis Comparison

Dual analysis is based on the fact that the system transfer matrix is directionally dependent. One end of the

system main member is designated as the starting point; then the component transfer matrices are multiplied together in sequence proceeding toward the other end. If an error were to occur in the structure of one of the transfer matrices or in the method of their combination to form the system transfer matrix,the results from starting at opposing ends could differ significantly.

An improperly constructed state matrix would also be evident from a dual analysis.

The straight pipe systems and branched systems showed no significant differences when subjected to dual analysis. The maximum difference observed in eighteen modes of the complex system dual analysis was 2.67%, while the average difference was 0.58%.

2. Solution Integrity

Solution integrity was established by observing that the mode frequencies detected by the program were, in fact, those of the system with no omissions. This was accomplished by comparison with analytical and reference results. Dual analysis comparison was also used in determining that the same modes had been detected in either direction. No discrepancies were noted in any of the systems checked on the basis of these comparisons.

Program VIBREL includes in its output a graph of frequency determinant versus frequency in order to check whether the frequency increment is such that a mode or modes have been skipped in the scanning process. In the

first analysis of a branched system which appears in Appendix F, the trend of the output plot showed that two natural frequencies in close proximity had been overlooked. When the program was rerun with the starting frequency increment decreased by a factor of 2, these two frequencies were detected.

The maximum number of natural frequencies which can be found using VIBREL is related to the significant figure capacity of the computer and the physical configuration of the piping system. As pointed out by Kim [Ref. 6], at the higher frequencies the columns of the frequency determinant approach the point of being parallel; hence the numerical value of the determinant includes differences of large numbers. Systems with few components generally exhibit this parallelism at a lower frequency than the more complex ones; however, there is no way of determining beforehand where numerical difficulties will be encountered. When parallelism is approached, a scattering of the values of frequency determinant will be noticed in the output plot.

Although the exponent limit on the IBM 360 computer used for testing the program is ± 78, program VIBREL has been coded to cease computation for a particular problem and print a message when the exponent of frequency determinant exceeds ± 60.

D. CONCLUSIONS AND RECOMMENDATIONS

Based on the results of the accuracy and integrity analyses previously discussed and through comparison with similar programs developed for planar cases, namely those of Fink [Ref 5] and Kim [Ref. 6], program VIBREL could be employed in its present form as a working tool for engineers in the dynamic analysis of spatial piping systems. The fact that eighteen mode frequencies were obtained for the complex system exhibited in Appendix F implies that at least the first few and usually the most useful frequencies can be determined for practical piping configuration with an accuracy consistent with engineering design.

Minor modifications for adapting the program to a particular situation can be made using the guidelines of Appendix C. With regard to major revisions, program VIBREL was developed with solution accuracy and integrity of first importance. It is recommended for future users that changes in the coding be made for a minimum computer execution time if that becomes a prime requisite. Avenues of investigation regarding numerical difficulties at higher frequencies are also open to the future user. A similar analysis for the planar case was undertaken by Kim [Ref. 6], who utilized alternatives to the frequency determinant approach.

Although the calculation of mode shapes of the piping system has not been included in the program because of time limitations, an outline of how to incorporate this feature is discussed in the following paragraphs. This modification

to the program is recommended as a further important step in the dynamic analysis of 3-D piping systems.

Let us consider again, for simplicity, the two-dimensional system of Fig. 2.3 which is repeated below. We found that the total system frequency condition could be expressed as:

$$\begin{bmatrix} a_{13} & a_{24} \\ a_{23} & a_{24} \end{bmatrix} \begin{Bmatrix} A_1 \\ A_2 \end{Bmatrix} = 0 \qquad (3\text{-}1)$$

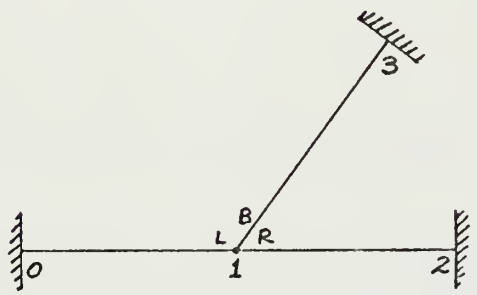


Figure 2.3: Single Branch Planar System

A vanishing determinant of the coefficient matrix yields a natural frequency. At this frequency there is a non-trivial solution to Eqs. 3-1 and A_1/A_2 can be found. Normalizing A_1 and A_2 gives the modal values of the non-zero quantities of the left end state vector, A_1^* and A_2^*. A suitable normalizing condition would be $A_1^2 + A_2^2 = 1$. With these values known we proceed along the main member calculating the modal deflections and forces at each divisional point. This is done simply by multiplying the normalized and compressed left end state vector by the system transfer matrix (2r x r). In Fig. 2.3 the modal forces and deflections to the left of point 1 would be

$$\begin{Bmatrix} w \\ \psi \\ v_z \\ M_z \end{Bmatrix}_{1b} = \begin{bmatrix} b_{13} & b_{14} \\ b_{22} & b_{24} \\ b_{33} & b_{34} \\ b_{43} & b_{44} \end{bmatrix} \begin{Bmatrix} A_1^* \\ A_2^* \end{Bmatrix} \tag{3-2}$$

In order to permit unequivocal interpretation of program output, it would be advisable to incorporate a point numbering scheme for each point at which modal forces and deflections are desired. In addition, these forces and deflections will be expressed in the local coordinate systems and a method should be incorporated for transforming them to the global system.

When a branch point, such as 1 in Fig. 2.3, is encountered, the branch transfer matrix up to point 1 is formed. Thus

$$\begin{Bmatrix} w \\ \psi \\ v_z \\ M_z \end{Bmatrix}_{1B} = \begin{bmatrix} c_{13} & c_{14} \\ c_{23} & c_{24} \\ c_{33} & c_{34} \\ 43 & c_{44} \end{bmatrix} \begin{Bmatrix} B_1 \\ B_2 \end{Bmatrix} \tag{3-3}$$

where B_1 and B_2 are the unknown quantities of the state vector at the remote end of the branch. Since we know that the deflections are continuous across the junction point

$$\begin{Bmatrix} w \\ \psi \end{Bmatrix}_{1B} = \begin{Bmatrix} w \\ \psi \end{Bmatrix}_{1L} \tag{3-4}$$

The quantities at 1L are given in terms of the known values A_1^* and A_2^* so that the values of B_1 and B_2 which are intrinsic to the 1B deflections may be solved for through the relations 3-4. The known values of B_1 and B_2 may now be used to find modal forces and deflections in a manner similar to that used for the main member.

Just to the right of the junction point, 1R, the modal deflections are the same as at the left; however, the forces must be calculated by adding the 1L forces to the 1B forces in the global coordinate system.

APPENDIX A

NOTATION AND NOMENCLATURE

The following matrix notation is used in the text:

[ ] - matrix

{ } - n x 1 column vector

DIAG { } - square matrix having the given elements in its principal diagonal and zero elsewhere

^{T} - superscript denoting the transpose of a given matrix

The following vector notation is used in the text:

$\bar{A}$ - vector A

$|\bar{A}|$ - length of vector A

$\bar{A} \cdot \bar{B}$ - dot product of vectors A and B

$\bar{A} \times \bar{B}$ - cross product of vectors A and B

Table A.1 first lists the symbols used in either the test or Appendix B, with the corresponding computer program variable identifiers. Following these appear the program variable names which have no counterpart in the text.

TABLE A.1

TEXT SYMBOL	PROGRAM VARIABLE	DESCRIPTION
A	AREA	cross sectional area of pipe
$\bar{A}$	A	first of two vectors defining a piping directional change

TABLE A.1 (Continued)

TEXT SYMBOL	PROGRAM VARIABLE	DESCRIPTION
B		translation matrix
$\bar{B}$	B	second of two vectors defining a piping directional change
C		compliance matrix
D		vector of generalized displacements
$\bar{e}_x$	UX	unit vector in local x-direction
E	EY	Young's modulus
F		vector of generalized forces
G	G	shear modulus
G_α	G2	branch coordinate transformation matrix
G_γ	G3	branch coordinate transformation matrix
$G_{\alpha 1}$	G1	branch coordinate transformation matrix
$\bar{i}$		unit vector in local x-direction
I_x	DIX	mass moment of inertia of a curved pipe about x-axis
$\bar{\bar{I}}_o$		inertia tensor about point O
I_6		6 x 6 identity matrix
i		radius of gyration of pipe cross section
$\bar{j}$		unit vector in local y-direction
J	DJ	moment of inertia of pipe section
$\bar{k}$		unit vector in local z-direction
k	BFF	bend flexibility factor
ℓ	DL	length of a straight section of pipe
L		rotation matrix
m	DM	mass of a section of pipe

TABLE A.1 (Continued)

TEXT SYMBOL	PROGRAM VARIABLE	DESCRIPTION
M_y		moment about local y-axis
M_z		moment about local z-axis
N		axial force in local x-direction
O_6		6 x 6 null matrix
P		point matrix
R	R	radius of curvature of pipe bend
$\bar{r}$	RBAR	mean radius of pipe section
R_1	R1	submatrix of branch transfer matrix
R_2	R2	submatrix of branch transfer matrix
r_G	RG	radius of center of mass of a curved section of pipe
$\tilde{s}_x$	SX	main member unit vector just prior to branch junction
t	T	pipe wall thickness
T		torque about local x-axis
u		deflection in x-direction
$\bar{\bar{u}}$		unit dyadic
$\tilde{U}_x$	UX	unit vector in local x-direction
$\tilde{U}_{x'}$	UXP	unit vector in local x'-direction
$\tilde{U}_{x''}$	UXDP	unit vector in local x"-direction
U	U, U1	transfer matrix
v		deflection in local y-direction
V		field matrix
V_y		shear force in local y-direction
V_z		shear force in local z-direction

TABLE A.1 (Continued)

TEST SYMBOL	PROGRAM VARIABLE	DESCRIPTION
w		deflection in local z-direction
z		state vector
α	AL	x-axis coordinate transformation angle
α_1	AL1	x-axis coordinate transformation angle
γ	GAM	z-axis coordinate transformation angle
Δ		frequency determinant
θ		angular deflection about z-axis
μ	UU	mass per unit length of pipe
δ	PR	Poisson's ratio
ξ	SDF	shear distribution factor
ρ	RHOM	mass density
ϕ		angular deflection about x-axis
ψ		angular deflection about y-axis
ω	W1,W2	circular frequency
	AA	array of section properties
	ARCL	arc length of curved pipe
	BL	distance from bend point to tangent point
	CTX12	12 x 12 x-axis coordinate transformation matrix
	CTZ12	12 x 12 z-axis coordinate transformation matrix
	D	value of frequency determinant
	D1	outside pipe diameter
	D2	inside pipe diameter

TABLE A.1 (Continued)

TEXT SYMBOL	PROGRAM VARIABLE	DESCRIPTION
	DD	frequency determinant array
	FN	array of natural frequencies
	GO	gravitational constant
	ISEC	section identifier code
	JBC	branch boundary condition code
	K	boundary condition sequence number
	KB	branch repeat discriminant
	LBC	left boundary condition code-main member
	LBR	branch point identifier discriminant
	M	dimension of array of frequency determinant values
	MBC	right boundary condition code-main member
	MCS	curved subsection override discriminant
	MSR	shear deflection/rotary inertia discriminant
	MST	straight section test discriminant
	N	section number
	NB	vector of boundary condition codes
	NIB	initial unit vector discriminant for a branch
	NID	point identifier code
	NIT	number of iterations required for Mueller's method
	NIV	initial unit vector discriminant for main member
	NK	dual branch discriminant
	NMO	number of modes requested in analysis

TABLE A.1 (Continued)

TEXT SYMBOL	PROGRAM VARIABLE	DESCRIPTION
	NN	iteration section number
	NPROB	number of systems to be analyzed by program
	NSEC	section number
	NSS	number of curved pipe subsections
	PX	X-coordinate of a working point
	PY	Y-coordinate of a working point
	PZ	Z-coordinate of a working point
	RHO	weight density
	UA	point matrix for a curved pipe
	UB,UCY	field matrix for a curved pipe
	UST	straight pipe transfer matrix
	WINCR	frequency increment
	WN	natural frequency
	X	array of frequencies for output plot
	Y	array of frequency determinant values for output plot

B.1 STRAIGHT PIPE TRANSFER MATRIX

$\cos\alpha_o$	0	0	0	0	0	$\frac{\ell}{EA}\frac{\sin\alpha_o}{\alpha_o}$	0	0	0	0	0	0
0	$C_o-\sigma C_2$	0	0	0	$\ell[C_1-(\sigma+\gamma)C_3]$	0	$-\frac{a\ell}{\beta^4}[-\sigma C_1+(\beta^4+\sigma^2)C_2]$	0	0	0	aC_2
0	0	$C_o-\sigma C_2$	0	$-\ell[C_1-(\sigma+\gamma)C_3]$	0	0	0	$-\frac{a\ell}{\beta^4}[-\sigma C_1+(\beta^4+\sigma^2)C_2]$	0	$-aC_2$	0
0	0	0	$\cos\gamma_o$	0	0	0	0	0	$\frac{\ell}{GJ}\frac{\sin\gamma_o}{\gamma_o}$	0	0
0	0	$-\frac{\beta^4}{\ell}C_3$	0	$C_o-\gamma C_2$	0	0	0	aC_2	0	$\frac{a}{\ell}(C_1-\gamma C_3)$	0
0	$\frac{\beta^4}{\ell}C_3$	0	0	0	$C_o-\gamma C_2$	0	$-aC_2$	0	0	0	$\frac{a}{\ell}(C_1-\gamma C_3)$
$-\mu\ell\omega^2\frac{\sin\alpha_o}{\alpha_o}$	0	0	0	0	0	$\cos\alpha_o$	0	0	0	0	0
0	$-\frac{\beta^4}{a\ell}(C_1-\sigma C_3)$	0	0	0	$-\frac{\beta^4}{a}C_2$	0	$C_o-\sigma C_2$	0	0	0	0
0	0	$-\frac{\beta^4}{a\ell}(C_1-\sigma C_3)$	0	$\frac{\beta^4}{a}C_2$	0	0	0	$C_o-\sigma C_2$	0	0	0
0	0	0	$-\mu\ell i_x^2\omega^2\frac{\sin\gamma_o}{\gamma_o}$	0	0	0	0	0	$\cos\gamma_o$	0	0
0	0	$-\frac{\beta^4}{a}C_2$	0	$\frac{\ell}{a}[-\gamma C_1+(\beta^4+\gamma^2)C_3]$	0	0	0	$\ell[C_1-(\sigma+\gamma)C_3]$	0	$C_o-\gamma C_2$	0
0	$\frac{\beta^4}{a}C_2$	0	0	0	$\frac{\ell}{a}[-\gamma C_1+(\beta^4+\gamma^2)C_3]$	0	$-\ell[C_1-(\sigma+\gamma)C_3]$	0	0	0	$C_o-\gamma C_2$

$\alpha_o = \ell\omega\sqrt{\frac{\mu}{AE}}$

$\gamma_o = \ell\omega\sqrt{\frac{\mu}{AG}}$

$a = \ell^2/EJ$

$\beta^4 = \mu\omega^2\ell^4/EJ$

$\sigma = \mu\omega^2\ell^2/GA\bar{F}$

$\tau = \frac{\ell^2}{E\sigma}(\mu i^2\omega^2)$

$\Lambda = \frac{1}{\lambda_1^4+\lambda_2^4}$

$\lambda_1 = \sqrt{\sqrt{\beta^4+\frac{1}{4}(\sigma-\gamma)^2}-\frac{1}{2}(\sigma+\gamma)}$

$\lambda_2 = \sqrt{\sqrt{\beta^4+\frac{1}{4}(\sigma-\gamma)^2}+\frac{1}{2}(\sigma+\gamma)}$

$C_o = \Lambda(\lambda_2^2\cosh\lambda_1+\lambda_1^2\cos\lambda_2)$

$C_1 = \Lambda(\lambda_2^2/\lambda_1\sinh\lambda_1+\lambda_1^2/\lambda_2\sin\lambda_2)$

$C_2 = \Lambda(\cosh\lambda_1-\cos\lambda_2)$

$C_3 = \Lambda(1/\lambda_1\sinh\lambda_1-1/\lambda_2\sin\lambda_2)$

B.2 CURVED PIPE FIELD MATRIX

$\cos\gamma$	$-\sin\gamma$	0	0	0	$R(\cos\gamma-1)$	$A_1\cos\gamma + A_3R(\cos\gamma-1) - A_2\sin\gamma$	$A_2\cos\gamma + \sin\frac{\gamma}{2}\cdot R\cdot(\cos\gamma-1) - A_2\sin\gamma$	0	0	0	$A_3\cos\gamma + GR(\cos\gamma-1) - H\sin\gamma\sin\frac{\gamma}{2}$
$\sin\gamma$	$\cos\gamma$	0	0	0	$R\sin\gamma$	$A_1\sin\gamma + A_3R\sin\gamma + A_2\cos\gamma$	$A_2\sin\gamma + HR\sin\gamma\sin\frac{\gamma}{2} + A_2\cos\gamma$	0	0	0	$A_3\sin\gamma + GR\sin\gamma + H\cos\gamma\sin\frac{\gamma}{2}$
0	0	1	$R(\cos\gamma-1)$	$-R\sin\gamma$	0	0	0	$C_1 + D_1R(\cos\gamma-1) - C_3R(\sin\gamma)$	$C_2 + D_2R(\cos\gamma-1) - D_2R\sin\gamma$	$C_3 + D_3R(\cos\gamma-1) - E_3R\sin\gamma$	0
0	0	0	$\cos\gamma$	$-\sin\gamma$	0	0	0	$D_1\cos\gamma - C_3\sin\gamma$	$D_2\cos\gamma - D_3\sin\gamma$	$D_3\cos\gamma - E_3\sin\gamma$	0
0	0	0	$\sin\gamma$	$\cos\gamma$	0	0	0	$D_1\sin\gamma + C_3\cos\gamma$	$D_2\sin\gamma + D_3\cos\gamma$	$D_3\sin\gamma + E_3\cos\gamma$	0
0	0	0	0	0	1	A_3	$H\sin\frac{\gamma}{2}$	0	0	0	G
0	0	0	0	0	0	$\cos\gamma$	$-\sin\gamma$	0	0	0	0
0	0	0	0	0	0	$\sin\gamma$	$\cos\gamma$	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	$R(\cos\gamma-1)$	$\cos\gamma$	$-\sin\gamma$	0
0	0	0	0	0	0	0	0	$R\sin\gamma$	$\sin\gamma$	$\cos\gamma$	0
0	0	0	0	0	0	$R(\cos\gamma-1)$	$-R\sin\gamma$	0	0	0	1

$\zeta = 2(1+\delta)\xi$

$k = 1.65\,\bar{F}^2/Rt$ (BUT ≥ 1)

$P = [R^2k + i^2(1-\zeta)]/2$

$a = (1+\delta-k)/2$

$\psi_1 = \gamma + \sin\gamma$

$\psi_2 = \gamma - \sin\gamma$

$A = (i^2\zeta\gamma + P\psi_1)R/EJ$

$B = (i^2\zeta\gamma + P\psi_2)R/EJ$

$M = [i^2\zeta + R^2(1+\delta)]\gamma R/EJ$

$D = [2R(1+\delta)\sin\frac{\gamma}{2}]R/EJ$

$E_1 = (k\gamma + Q\psi_1)R/EJ$

$F = (k\gamma + Q\psi_2)R/EJ$

$G = k\gamma R/EJ$

$H = -2R^2k\sin\frac{\gamma}{2}/EJ$

$A_1 = A\cos^2\frac{\gamma}{2} + 2RH\cos\frac{\gamma}{2} + B\sin^2\frac{\gamma}{2} + R^2G$

$A_2 = (B-A)\sin\gamma/2 - RH\sin\frac{\gamma}{2}$

$A_3 = -H\cos\frac{\gamma}{2} - RG$

$C_1 = M - 2RD\cos\frac{\gamma}{2} + R^2E_1\cos^2\frac{\gamma}{2} + R^2F\sin^2\frac{\gamma}{2}$

$C_2 = -D\cos\frac{\gamma}{2} + RE_1\cos^2\frac{\gamma}{2} + RF\sin^2\frac{\gamma}{2}$

$C_3 = D\sin\frac{\gamma}{2} + R\sin\gamma(F-E_1)/2$

$D_1 = -D\cos\frac{\gamma}{2} + RE_1\cos^2\frac{\gamma}{2} + RF\sin^2\frac{\gamma}{2}$

$D_2 = E_1\cos^2\frac{\gamma}{2} + F\sin^2\frac{\gamma}{2}$

$D_3 = \sin\gamma(F-E_1)/2$

$E_3 = E_1\sin^2\frac{\gamma}{2} + F\cos^2\frac{\gamma}{2}$

B.3 CURVED PIPE POINT MATRIX

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-m\omega^2 & 0 & 0 & 0 & 0 & -m\omega^2\bar{y} & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -m\omega^2 & 0 & 0 & 0 & -m\omega^2\bar{x} & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -m\omega^2 & m\omega^2\bar{y} & m\omega^2\bar{x} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -I_{xo}\omega^2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -I_{yo}\omega^2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -I_{zo}\omega^2 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

$r_c = R \sin \gamma/2 / \gamma/2$

$\bar{x} = r_c \sin \gamma/2$

$\bar{y} = R - r_c \cos \gamma/2$

$$I_{xo} = \frac{\rho A R F^2}{2}\left[\frac{3\gamma}{2} + \frac{\sin 2\gamma}{4}\right] + 2\rho A R^3\left[\frac{3\gamma}{4} - \sin\gamma + \frac{\sin 2\gamma}{8}\right]$$

$$I_{yo} = \frac{\rho A R F^2}{2}\left[\frac{3\gamma}{2} - \frac{\sin 2\gamma}{4}\right] + 2\rho A R^3\left[\frac{\gamma}{4} - \frac{\sin 2\gamma}{8}\right]$$

$$I_{zo} = \frac{\rho A R F^2}{2}\gamma + 2\rho A R^3(\gamma - \sin\gamma)$$

APPENDIX C

DESCRIPTION OF PROGRAM

1. General Remarks

Program VIBREL is a double precision FORTRAN language digital computer program which is capable of performing a frequency analysis of a three-dimensional piping system. The program accomplishes this analysis by means of the transfer matrix method in which each section of pipe is characterized by a matrix known as a transfer matrix. These matrices which are functions of frequency are multiplied together in succession to form a transfer matrix which is characteristic of the entire system. To combine satisfactory accuracy with minimum computer time, a lumped mass model is used for curved sections of pipe while for straight pipe, a distributed mass approach is employed. Shear deflection and rotary inertia are optional in the model.

The solution is obtained by incrementing the frequency and forming the system transfer matrix and frequency determinant with each increment. The sign of the frequency determinant is used to control the search for and convergence to system natural frequencies. The convergence acceptability criterion may be set arbitrarily by the user depending on the accuracy desired. (See section 3.b of this appendix.) As one would expect, the greater the accuracy, the more computer time required.

The system is defined by working points which are read into the program with the intensive and extensive properties of the section of pipe preceding it. The details for this procedure are contained in Appendix E. The intensive and extensive properties may not vary along a single section of pipe but may vary from section to section. Projective complexities, including branches emanating from branches, cannot be handled directly by the program.

Three boundary conditions, fixed, free, or ball joint, may be specified by coded input for the ends of the main member and the remote ends of the branches. Alternative end conditions are discussed in section 3.b of this appendix.

2. Program Structure

a. Main Program

A simplified flow chart is shown in Fig. C.1. This discussion will pertain to that diagram.

Data which contains working points, intensive and extensive properties, and boundary condition codes is read into the program as it is required for computation. Local coordinate systems are set up at the starting point, prior to and following each directional change, and at branch junction points. From the working point locations and unit vectors describing the local coordinate systems, section lengths and coordinate rotation angles are computed.

For computational purposes piping bends are subsectioned based on the length-to-diameter ratio in the following

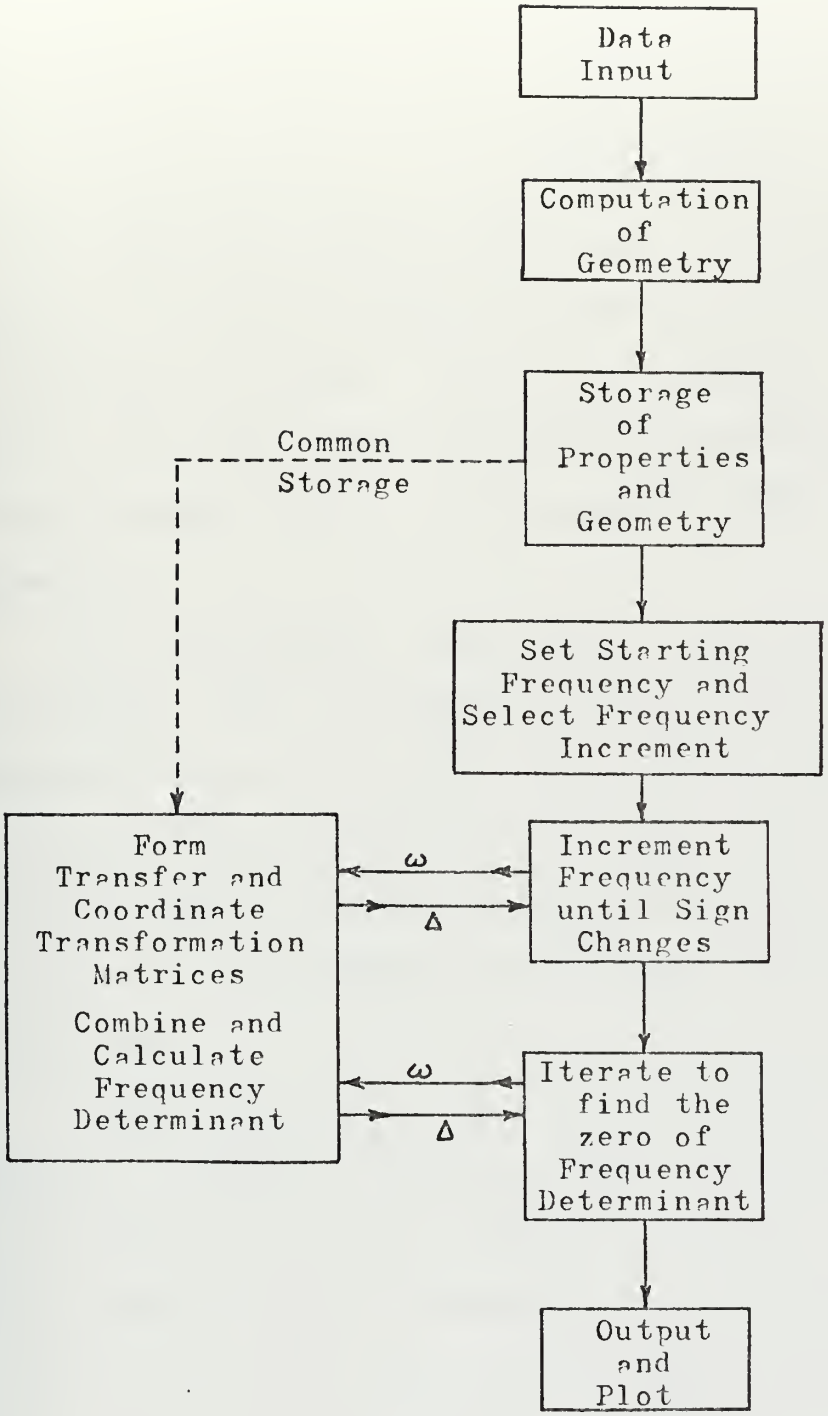


Figure C.1: Simplified Flow Diagram for Main Program

manner:

$0 < L/D \leq 1$	1 SUBSECTION
$1 < L/D \leq 3$	2 SUBSECTIONS
$3 < L/D \leq 6$	3 SUBSECTIONS

If L/D is greater than six, the number of subsections is computed as twice the number of mode frequencies sought, up to a maximum of twelve. The user may, however, choose to override the number of subsections as determined by the program through the use of the parameter MCS described in Appendix E.

As the lengths, subsections, and rotation angles are computed, they are stored, together with intensive and extensive properties, in an array. This array is passed via common storage (dotted line in Fig. C.1) to function subroutine FRDET (FRequency DETerminant) which later forms and evaluates the frequency determinant for a given value of frequency.

When the geometry for the entire system has been computed, frequency iteration begins. The starting frequency is arbitrarily set at 0.1 radians per second. To compute the starting frequency increment, the program constructs a synthetic straight pipe, with a length equal to the total length of the main member, and having outside diameter, wall thickness, and intensive properties equal in magnitude to a weighted average of the main member sections. One sixth of the fundamental frequency of this synthetic pipe

with fixed ends is taken as the starting frequency increment. The system transfer matrix and frequency determinant are formed by function subroutine FRDET for each new frequency. When a sign change is detected in the frequency determinant, an iteration process using Mueller's method of successive bisections and inverse parabolic interpolation is used for convergence to the natural frequency. The convergence acceptability criterion is taken as the starting frequency increment divided by ten thousand.

After a root has been located, the search begins anew with the starting frequency equal to the value of the natural frequency plus one radian per second. When the second mode frequency has been located, the frequency increment is changed to one tenth the difference between the previous two natural frequencies.

The search process continues until the specified number of frequencies has been found or until the significant figure capacity of the computer has been exceeded. Then the mode frequencies are printed and the pairs of frequencies and frequency determinant values used in the search are sorted and plotted.

b. Subroutines

The program utilizes sixteen subroutines in the process of computation. A brief description follows:

ANGLE	Computes the sign and the angular separation of two sets of unit vectors which have a common axis.

COORDX	Forms an n x n (n is some multiple of 3 and $\leq$ 12) coordinate transformation matrix for a rotation about a local x-axis.
COORDZ	Forms an n x n coordinate transformation matrix for a rotation about a local z axis.
CURMAT	Constructs the field matrix for a curved section of pipe.
DETER	Computes the value of an n x n determinant; used to evaluate the frequency determinant.
DROOT	Performs successive bisections and inverse parabolic interpolation to locate the zeros of the frequency determinant after a sign change has been detected.
INVERT	Inverts the matrix R_1 to be used in computing the branch point matrix.
MATMUL	Multiplies two matrices together. One of the matrices must be square.
POINT	Constructs a point matrix for a curved section of pipe.
PRPLOT	Takes the array of frequencies and frequency determinant values calculated during iteration and plots them lengthwise with the printer. Scaling is done automatically and the length of plot may be changed to suit the user.
SORT	Sorts the array of frequencies and frequency determinant values from iteration for plotting on the printer.
STATEM	Constructs the state matrix for the appropriate fixed, free, or ball joint boundary conditions.
STRMAT	Constructs the transfer matrix for a straight section of pipe.
UVEC	Computes a set of orthogonal unit vectors from two pipe section vectors which represent a directional change in the piping.

WSTART	Computes the value of the frequency determinant for a previously constructed synthetic pipe equal in length to the main member and having weighted average values of the properties of the main member sections. This frequency value is used in selecting the starting frequency increment.
FRDET	For a particular value of frequency, this subroutine assembles the state matrices, transfer matrices for each section of pipe, and the coordinate transformation matrices. It then multiplies them together in sequence and calculates the frequency determinant.

An interesting phenomenon occurs in the formation of the frequency determinant for a single straight pipe. Because of the symmetry of a pipe in the y and z directions, the two flexural submatrices in the straight pipe transfer matrix are identical, which causes the frequency determinant to maintain the same sign on either side of the flexural natural frequencies. This isolated case is handled in subroutine STRMAT through the use of the straight section test discriminant and appropriate exclusion of one of the flexural submatrices.

3. Remarks on Instructions for Program Use

a. General Remarks

Detailed instructions for program use are contained in the listing of Appendix E. It was intended for these instructions to be part of the program in order that any user, given only the program listing or the computer card deck, could employ the VIBREL package with no prior knowledge of transfer matrices and without perusal of this thesis.

Some clarification of the instructions for program use is justified for cases where properties or geometry change within a piping bend. First, consider the simple section of piping shown in Fig. C.2. Assuming that no change

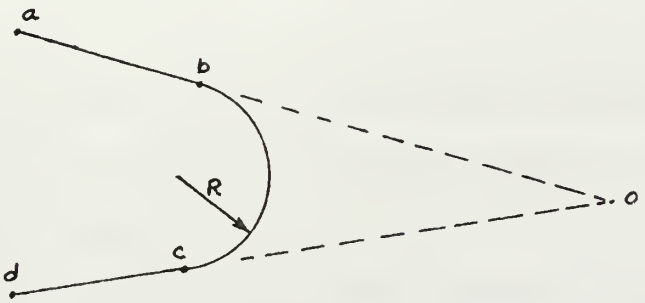


Figure C.2: Piping Bend with No Change in Properties

in cross section or properties occurs between points a and d, the only points which must be listed as working points for program input are a, o, and d. Using these points and the radius of curvature, program VIBREL will calculate lengths ab and cd as well as the arc length and angle of the bend.

Now consider the section of piping of Fig. C.3 where the cross section or properties change at point c but remain

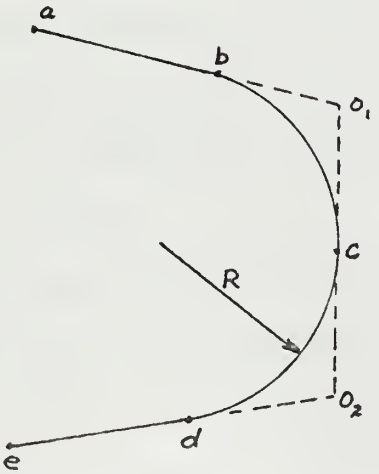


Figure C.3: Piping Bend with Change of Properties

constant between points a and c and between points c and e. In this instance points a, o_1, o_2, and e must be listed as working points for the program. VIBREL calculates straight lengths ab and de as well as the information required for curved lengths bc and cd. It should be obvious from this example that for the general case of two curved sections occurring sequentially that the intersection point (point c in Fig. C.3) need not be listed.

If the geometry input is such that negative lengths are computed, an error message is printed. One half inch leeway is allowed for negative lengths to take into account any errors which may occur in measurement.

b. Program Flexibilities

(1) Approximating Small Piping Accessories

Small valves, flanges, couplings and other piping accessories whose center of mass is relatively close to the centroidal axis of the pipe can be approximated using a straight section of pipe. The length, mass density, diameter and wall thickness can be artificially varied to give the mass of the item. The resulting variation in stiffness between an actual flange and artificial pipe, for example, usually will not have a significant effect on the results because of its short length.

(2) Approximating Alternative End Conditions

When the ideal boundary conditions are applied to the state vector on the end of a system, it must have six zero and six non-zero elements. Actual field conditions

may dictate that this is not the case because of flexible mountings or connection to machinery. A good approximation in these circumstances may be made by attaching one end of an artificial straight pipe to the system piping keeping its other end fixed. The mass of the artificial pipe should be small with the extensive properties and elastic modulus chosen to approximate the stiffness of the mounting.

The program can then use the ideal boundary conditions with little loss of accuracy from the real situation.

(3) Program Modifications

In some instances, a user may desire to change such quantities as frequency increment or input format to suit his particular needs. For this reason, Table C.1 lists the more important program variables and what card or cards to make these changes to.

TABLE C.1

Modification	Card Numbers	Variables Affected
Maximum number of points for print plot	2890	x(400), y(400)
Maximum number of piping sections in analysis	2910	AA(100,10)
Maximum number of branches in analysis	2920	NB(50)
Input format	3210-50	
Output format (Input Data)	3390-420	
Output format (Properties and Geometry)	7750-60	
Output format (Natural Frequencies)	8740	

TABLE C.1 (Continued)

Modification	Card Number	Variables Affected
Starting frequency increment	8650	WINCR
Starting frequency	8760	W1
Convergence acceptability criterion	9120	EPS
Maximum iterations to convergence in DROOT	3670	NIT
Starting Frequency after location of natural frequency	9330	W1
Frequency increment after second mode	9320	WINCR
Number of lines of print plot	14730	RY

APPENDIX D

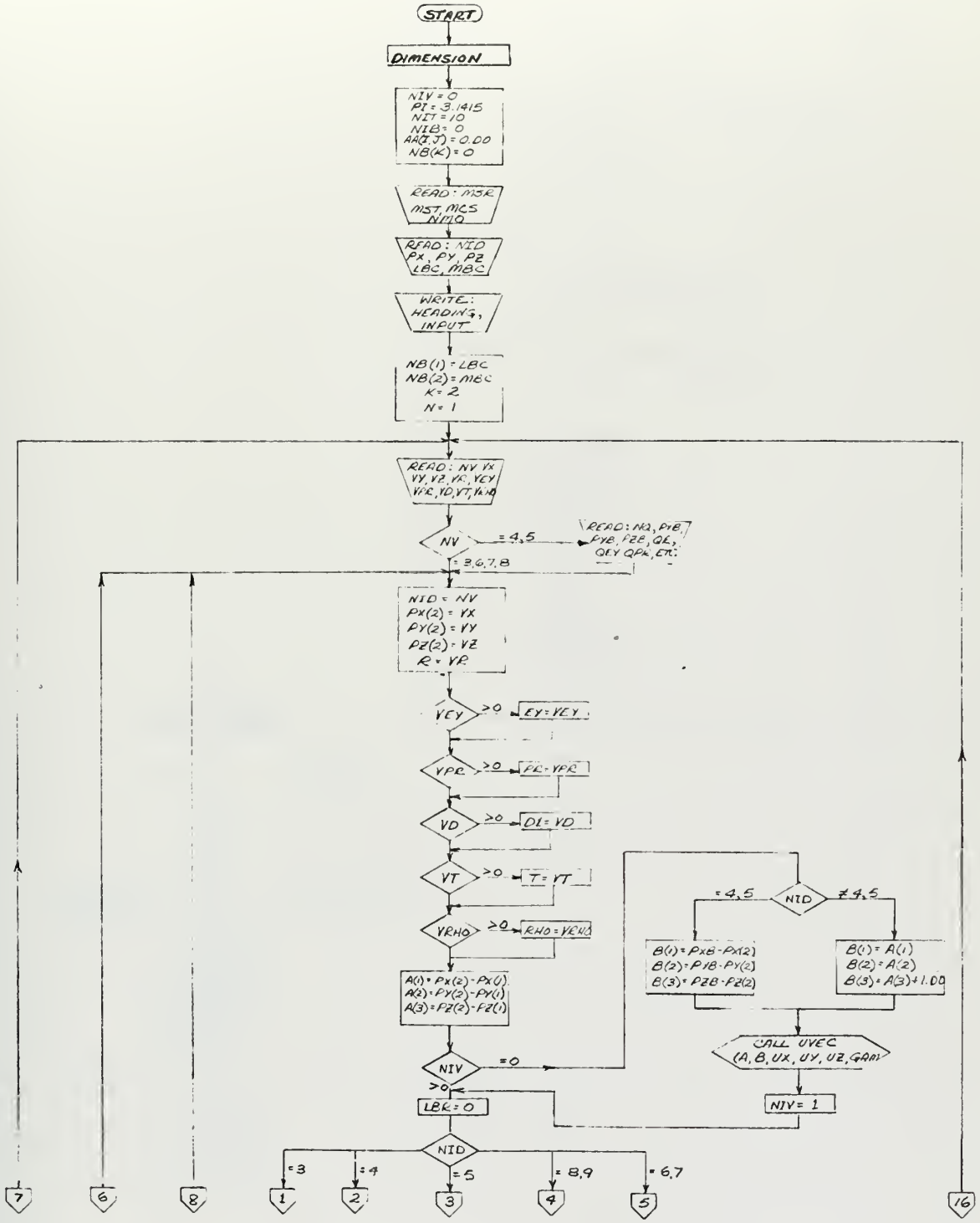
PROGRAM FLOW DIAGRAMS

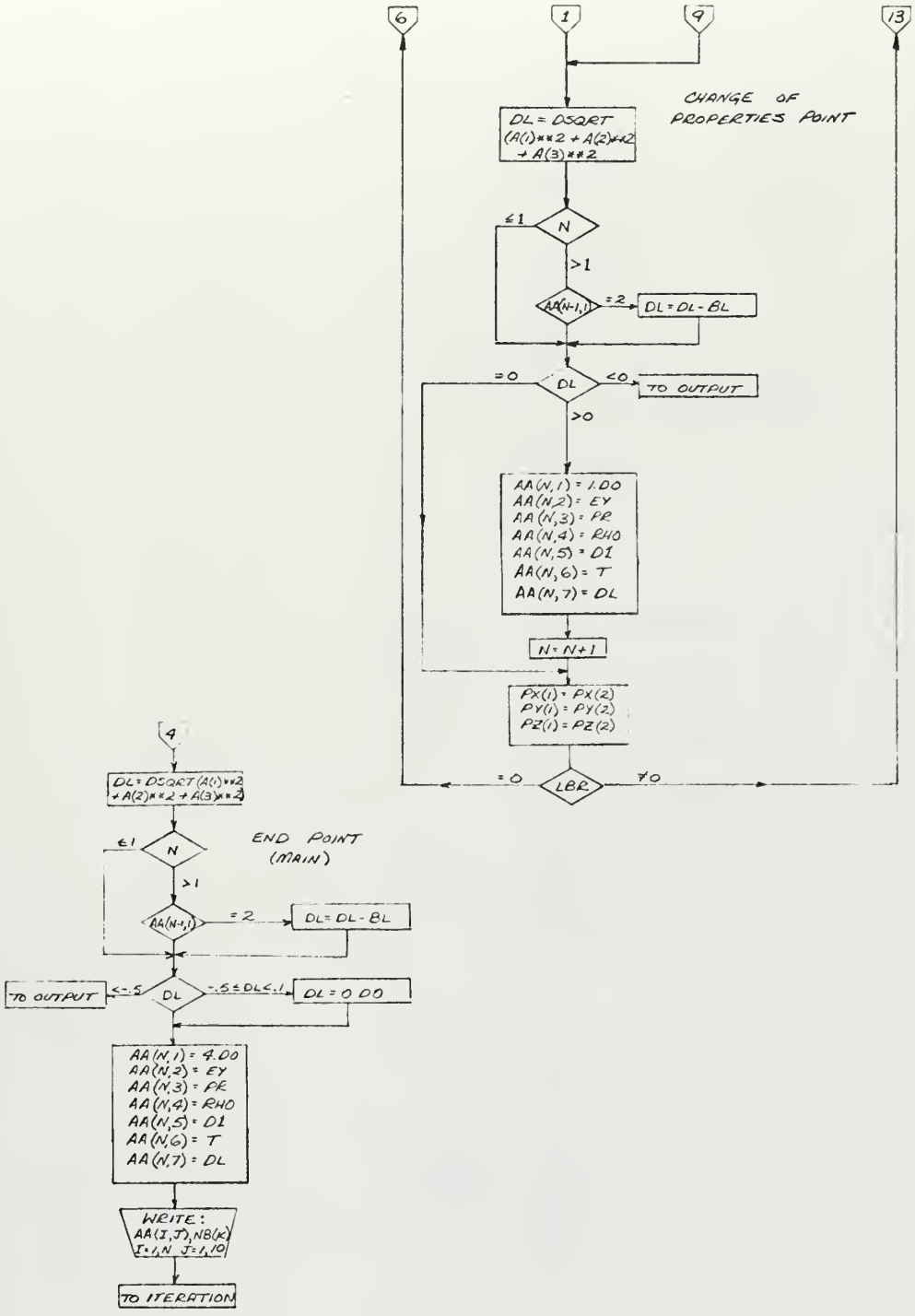
1. General Remarks

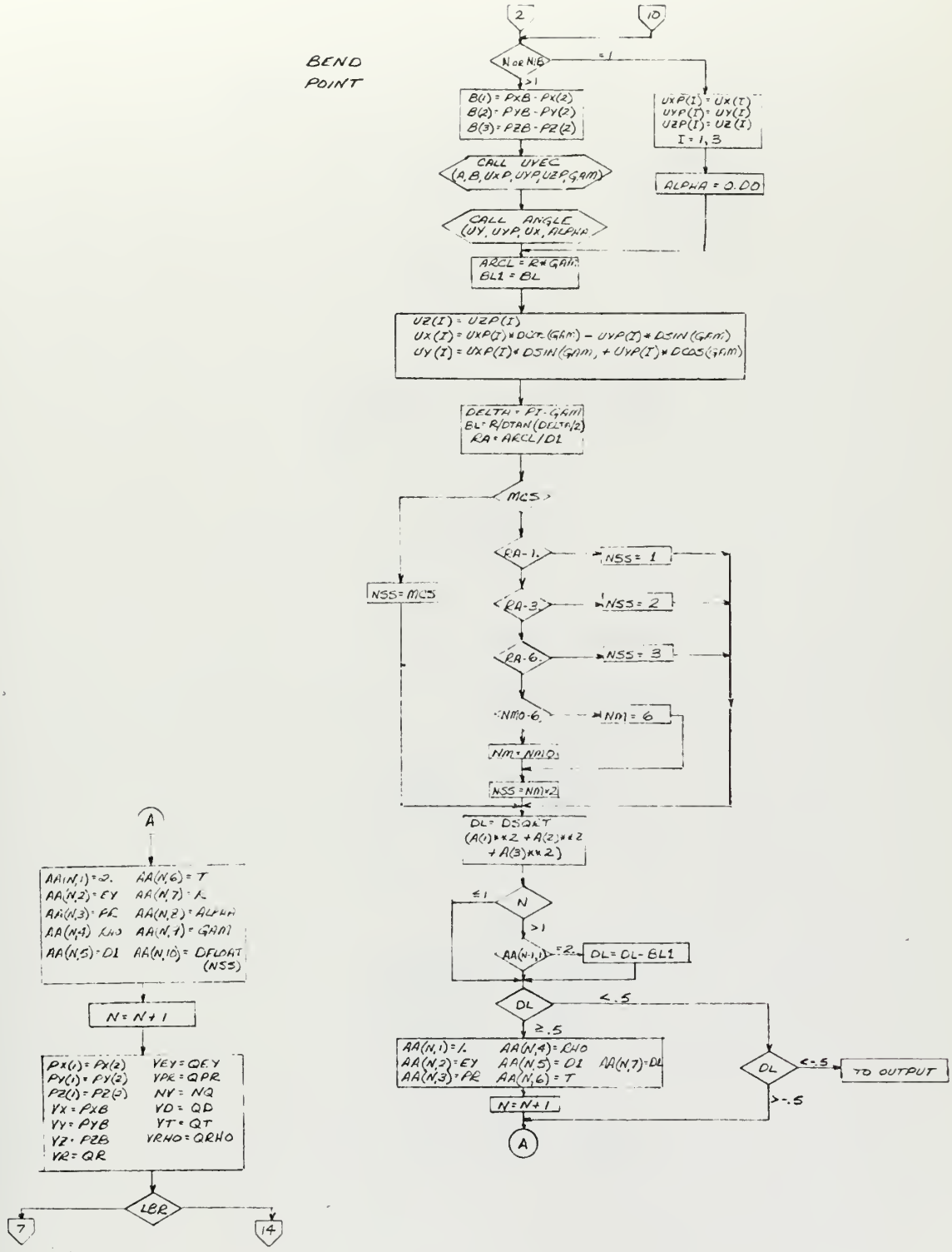
Flow diagrams are included for the main program and function subroutine FRDET. Subroutines DETER, DROOT, INVERT, and PRPLOT are standard package subroutines with minor modifications made to accommodate them to program VIBREL. Since their equivalent or an alternative routine may be substututed for them, flow diagrams are not included. Adequate comment statements are, however, contained in these subroutines to define their structure. Other subroutines for which no flow diagrams appear, incorporate, at most, two decisions in their structure.

2. Main Program Flow Diagram

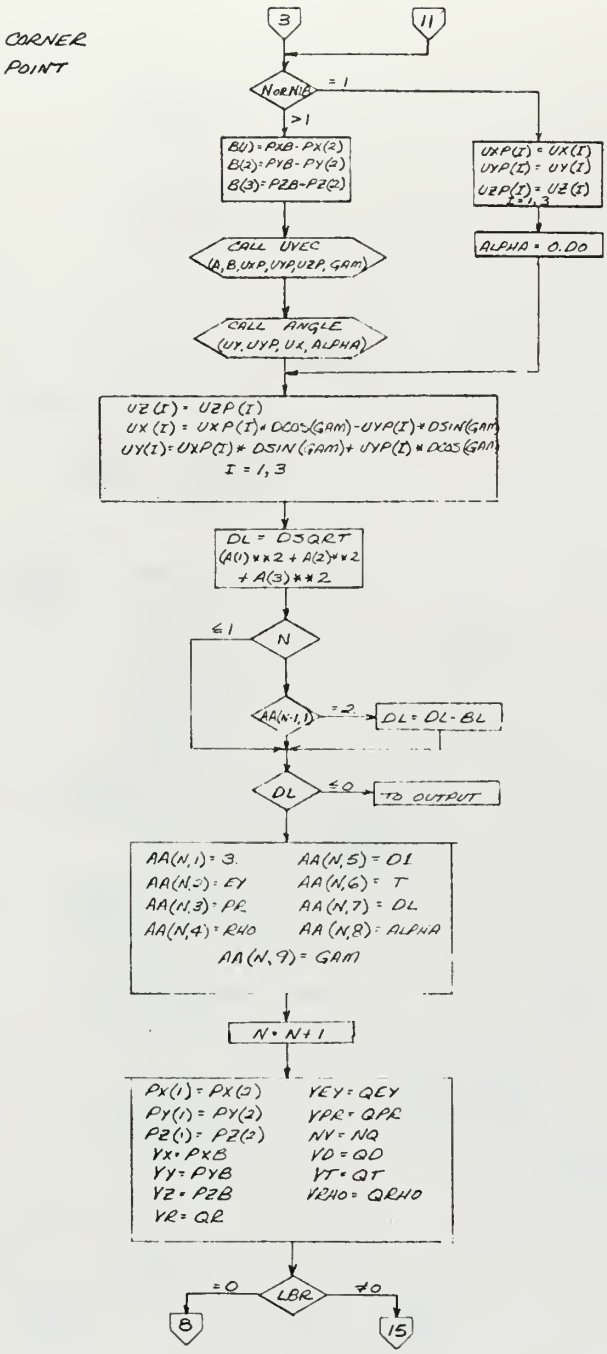
3. Function Subroutine FRDET Flow Diagram

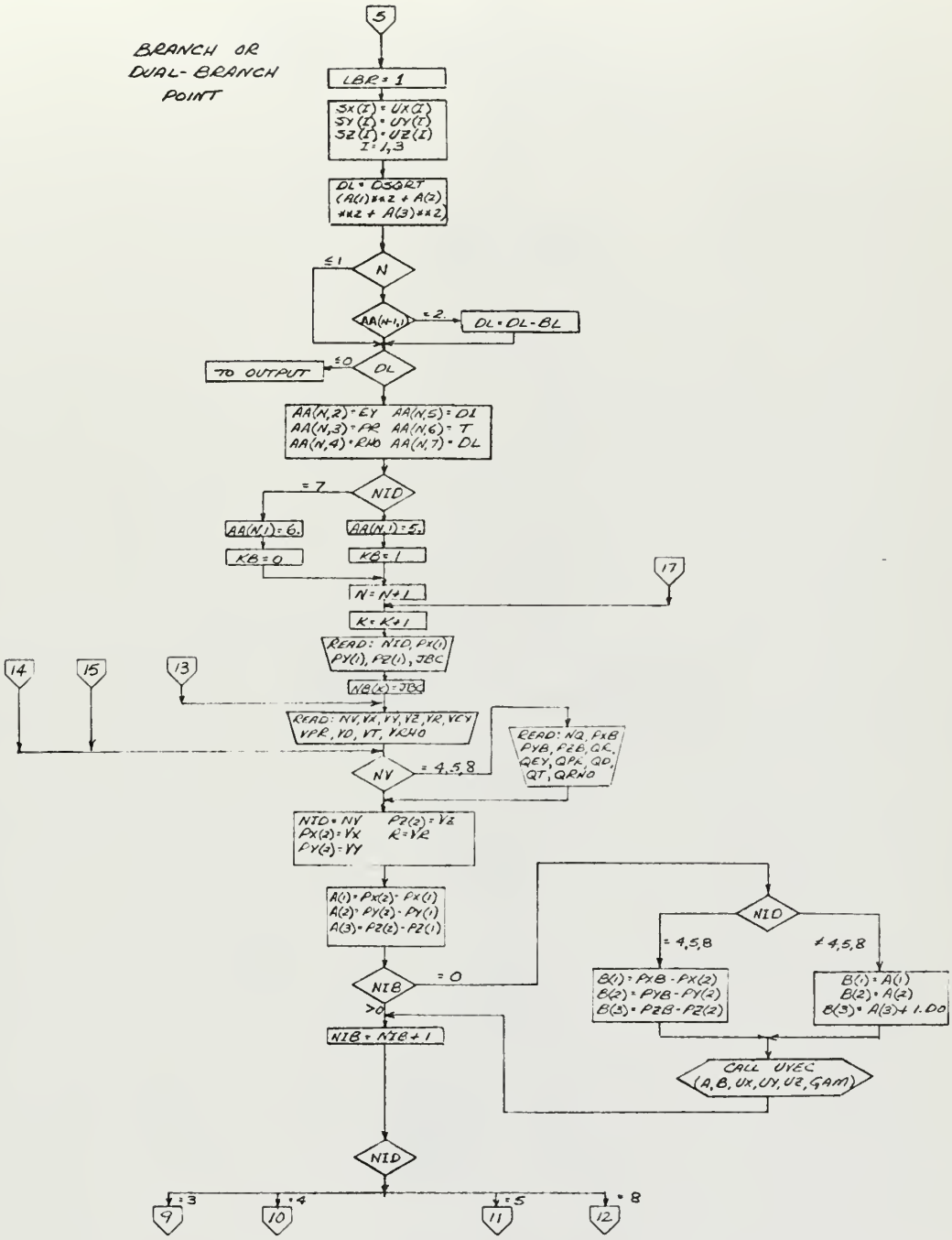

START
DIMENSION
NIV = 0
PI = 3.1415
NIT = 10
NIB = 0
AA(I,J) = 0.00
NB(K) = 0
READ: MSR
MST, MCS
NMO
READ: NID
PX, PY, PZ
LBC, MBC
WRITE:
HEADING,
INPUT
NB(1) = LBC
NB(2) = MBC
K = 2
N = 1
READ: NV VX
VY, VZ, VR, VEY
VER, VD, VT, VRHO
NV
= 4,5
READ: NA, PYB,
PYB, PZB, QL,
QEY QPR, ETC.
= 3,6,7,8
NID = NV
PX(2) = VX
PY(2) = VY
PZ(2) = VZ
R = VR
VEY
>0
EY = VEY
VPR
>0
ER = VPR
VD
>0
D1 = VD
VT
>0
T = VT
VRHO
≥0
RHO = VRHO
A(1) = PX(2) - PX(1)
A(2) = PY(2) - PY(1)
A(3) = PZ(2) - PZ(1)
NIV
=0
>0
LBR = 0
NID
= 3
= 4
= 5
= 8,9
= 6,7
NID
= 4,5
≠ 4,5
B(1) = PXB - PX(2)
B(2) = PYB - PY(2)
B(3) = PZB - PZ(2)
B(1) = A(1)
B(2) = A(2)
B(3) = A(3) + 1.00
CALL UVEC
(A, B, UX, UY, UZ, GAM)
NIV = 1
7
6
8
1
2
3
4
5
16

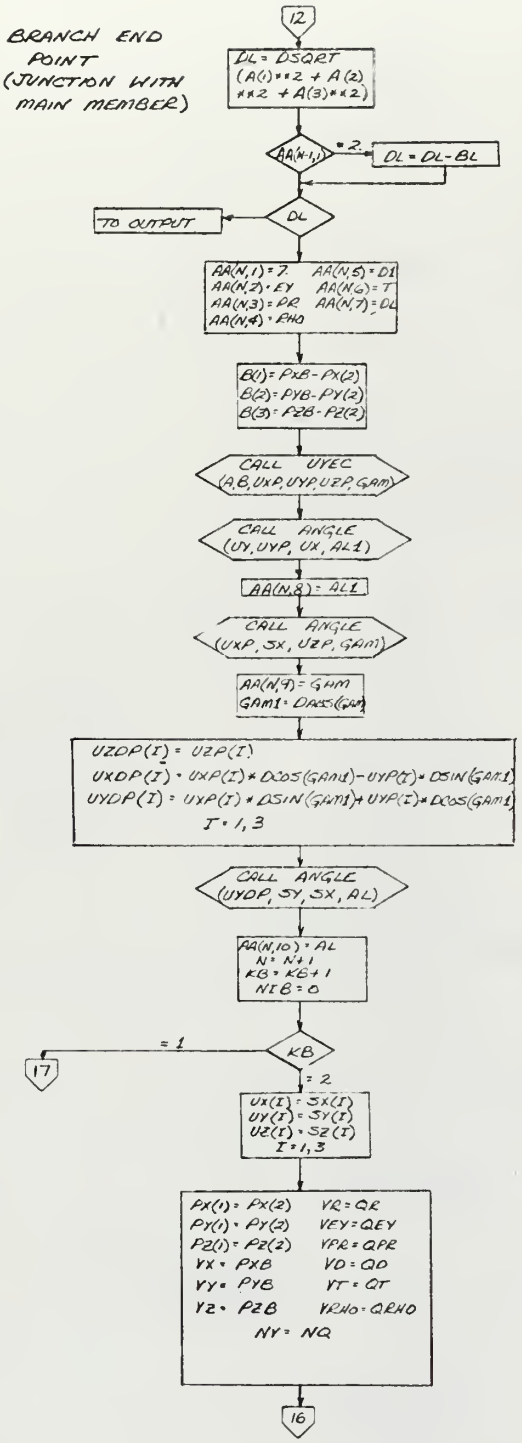

6
1
9
13
CHANGE OF PROPERTIES POINT
DL = DSQRT (A(1)**2 + A(2)**2 + A(3)**2
N
<1
>1
AA(N-1,1)
=2
DL = DL - BL
DL
=0
<0
TO OUTPUT
>0
AA(N,1) = 1.00
AA(N,2) = EY
AA(N,3) = PR
AA(N,4) = RHO
AA(N,5) = D1
AA(N,6) = T
AA(N,7) = DL
N = N+1
PX(1) = PX(2)
PY(1) = PY(2)
PZ(1) = PZ(2)
LBR
=0
≠0
4
DL = DSQRT (A(1)**2 + A(2)**2 + A(3)**2)
END POINT (MAIN)
<=-.5
-.5 ≤ DL ≤ .1
DL = 0.00
AA(N,1) = 4.00
AA(N,2) = EY
AA(N,3) = PR
AA(N,4) = RHO
AA(N,5) = D1
AA(N,6) = T
AA(N,7) = DL
WRITE: AA(I,J), NB(K) I=1,N J=1,10
TO ITERATION

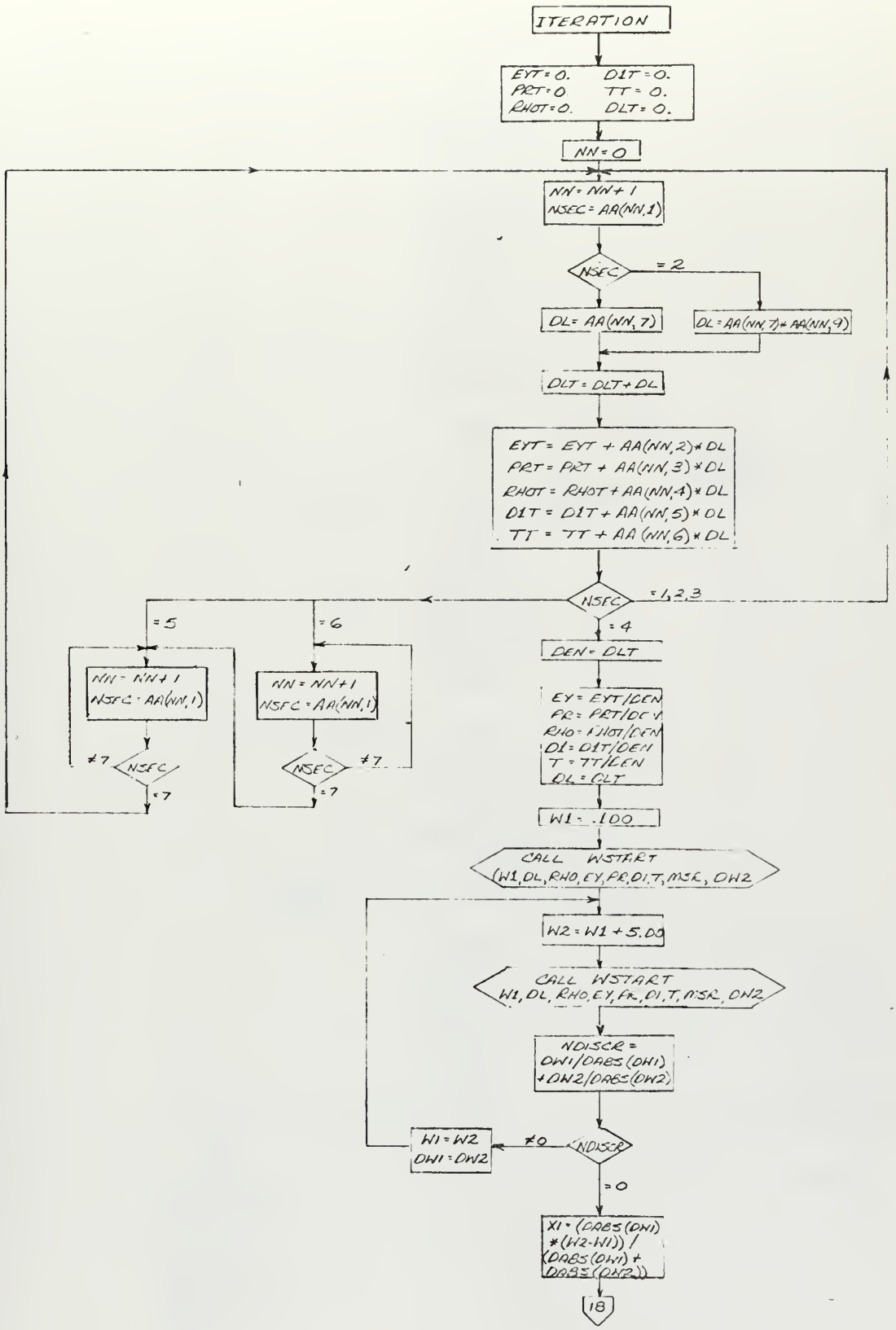

BEND POINT
2
10
N OR NB
=1
>1
B(1) = PXB - PX(2)
B(2) = PYB - PY(2)
B(3) = PZB - PZ(2)
UXP(I) = UX(I)
UYP(I) = UY(I)
UZP(I) = UZ(I)
I = 1, 3
CALL UVEC (A, B, UXP, UYP, UZP, G, AM)
ALPHA = 0.00
CALL ANGLE (UY, UYP, UX, ALPHA)
ARCL = R * GAM
BL1 = BL
UZ(I) = UZP(I)
UX(I) = UXP(I) * DCOS(GAM) - UYP(I) * DSIN(GAM)
UY(I) = UXP(I) * DSIN(GAM) + UYP(I) * DCOS(GAM)
DELTH = PI - GAM
BL = R/DTAN(DELTH/2)
RA = ARCL/D1
MCS
RA - 1.
NSS = 1
NSS = MCS
RA - 3.
NSS = 2
RA - 6.
NSS = 3
NM10 - 6.
NM = 6
NM = NM10
NSS = NM * 2
DL = DSQRT (A(1)**2 + A(2)**2 + A(3)**2)
N
≤1
>1
AA(N-1,1) = 2.
DL = DL - BL1
DL
<.5
≥.5
AA(N,1) = 1.
AA(N,2) = EY
AA(N,3) = PR
AA(N,4) = RHO
AA(N,5) = D1
AA(N,6) = T
AA(N,7) = DL
N = N + 1
A
AA(N,1) = 0.
AA(N,2) = EY
AA(N,3) = PL
AA(N,4) = RHO
AA(N,5) = D1
AA(N,6) = T
AA(N,7) = L
AA(N,8) = ALPHA
AA(N,9) = GAM
AA(N,10) = DFLOAT (NSS)
N = N + 1
PX(1) = PX(2)
PY(1) = PY(2)
PZ(1) = PZ(2)
VX = PXB
VY = PYB
VZ = PZB
VR = QR
YEY = QEY
YPR = QPR
NV = NQ
VD = QD
VT = QT
VRHO = QRHO
LBR
7
14
DL
<-.5
>-.5
TO OUTPUT

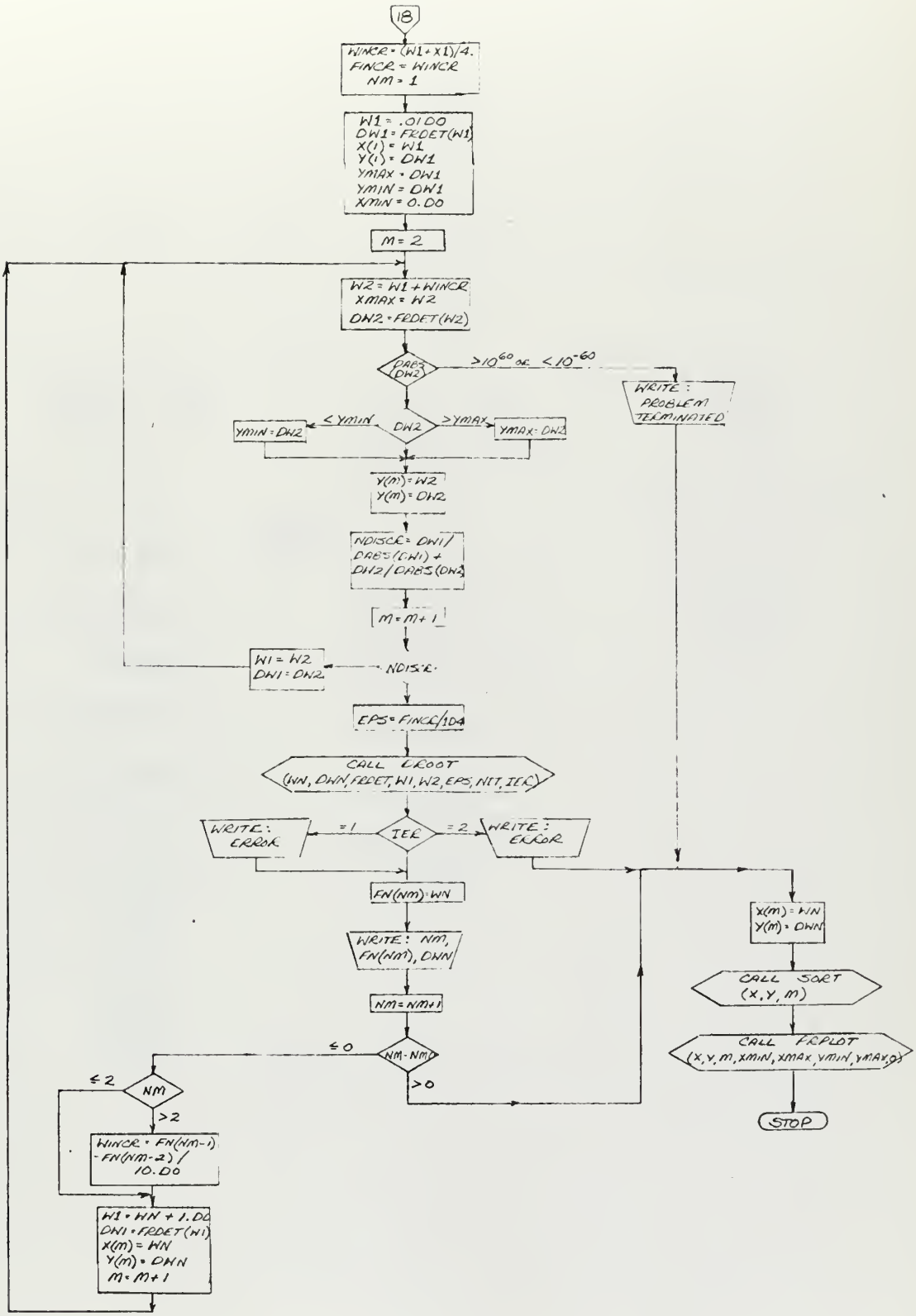
CORNER POINT

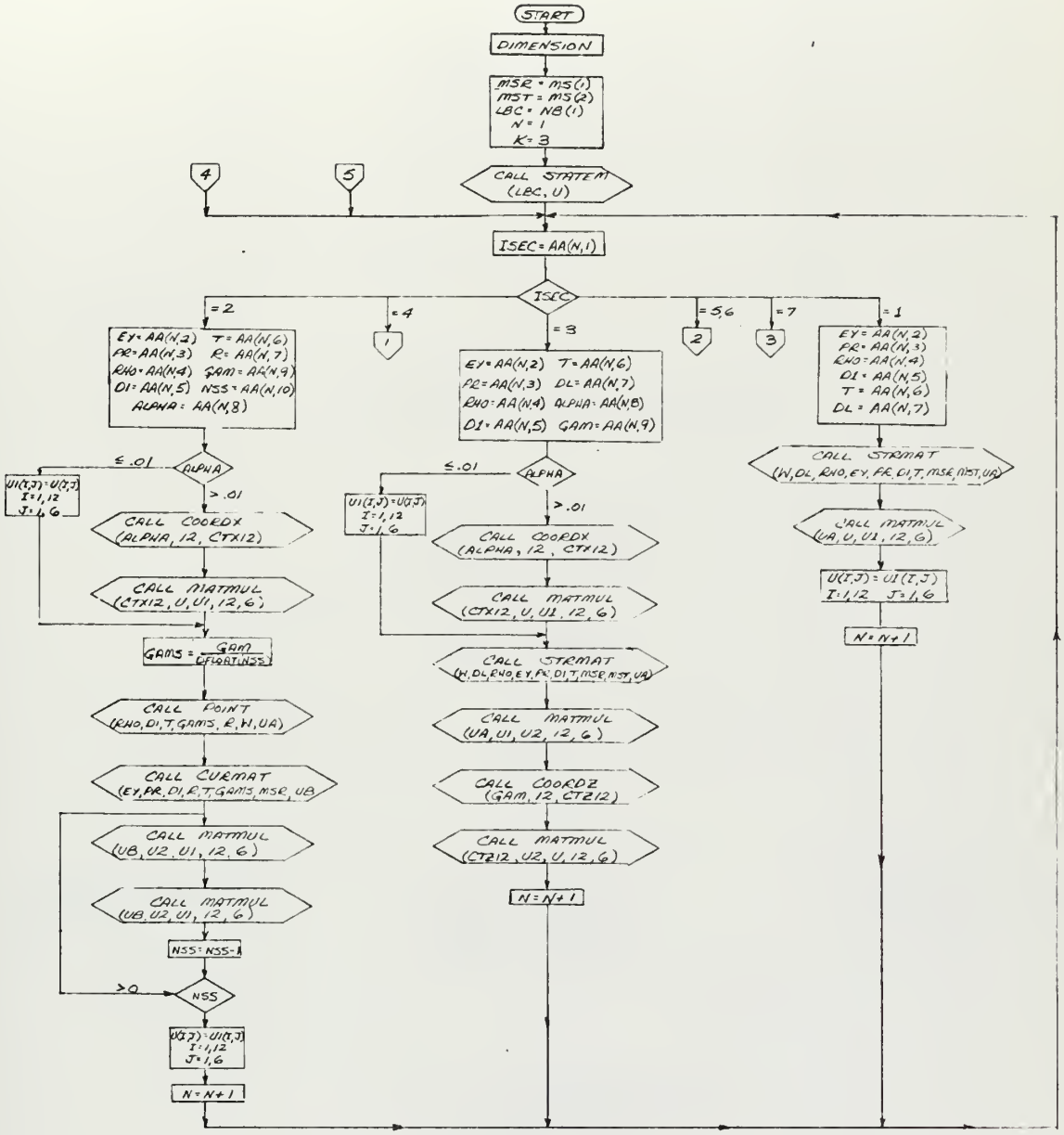


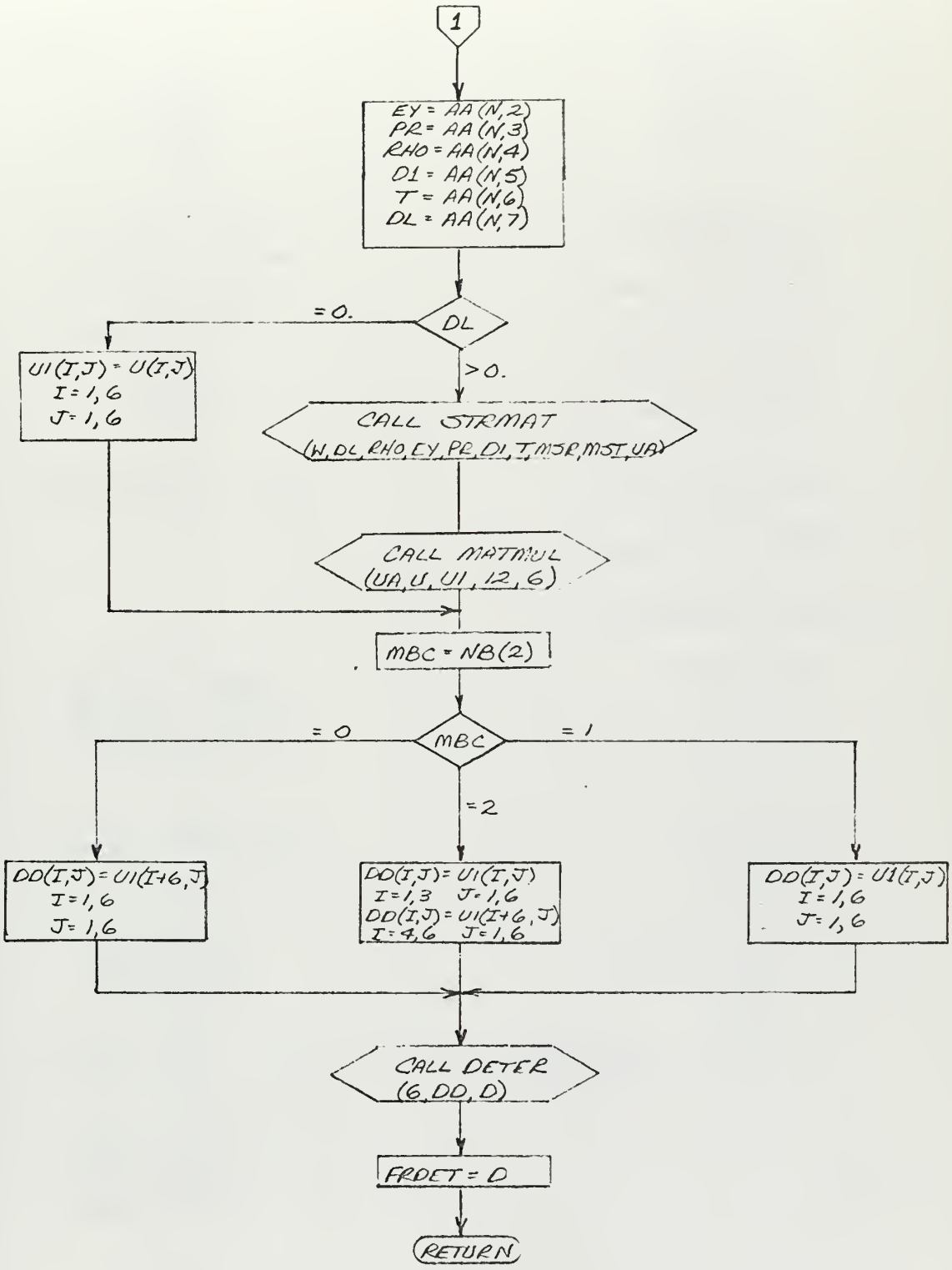

BRANCH OR DUAL-BRANCH POINT
5
LBR = 1
SX(I) = UX(I)
SY(I) = UY(I)
SZ(I) = UZ(I)
I = 1,3
DL = DSQRT (A(1)**2 + A(2) **2 + A(3)**2)
N
≤1
AA(N+1,1)
=2.
DL = DL - BL
DL
≤0
TO OUTPUT
AA(N,2) = EY AA(N,5) = DL
AA(N,3) = PR AA(N,6) = T
AA(N,4) = RHO AA(N,7) = DL
NID
=7
AA(N,1) = 6.
KB = 0
AA(N,1) = 5.
KB = 1
17
N = N+1
K = K+1
READ: NID, PX(1), PY(1), PZ(1), JBC
14
15
13
NB(K) = JBC
READ: NV, VX, VY, VZ, VR, VEY, VPR, VO, VT, VCHO
READ: NQ, PXB, PYB, PZB, QE, QEY, QPR, QO, QT, QRNO
NV
= 4,5,8
NID = NV PZ(2) = VZ
PX(2) = VX R = VR
PY(2) = VY
A(1) = PX(2) - PX(1)
A(2) = PY(2) - PY(1)
A(3) = PZ(2) - PZ(1)
NID
= 4,5,8
≠ 4,5,8
B(1) = PXB - PX(2)
B(2) = PYB - PY(2)
B(3) = PZB - PZ(2)
B(1) = A(1)
B(2) = A(2)
B(3) = A(3) + 1.00
CALL UVEC (A, B, UX, UY, UZ, GAM)
NIB
= 0
>0
NIB = NIB + 1
NID
=3
=4
=5
=8
9
10
11
12

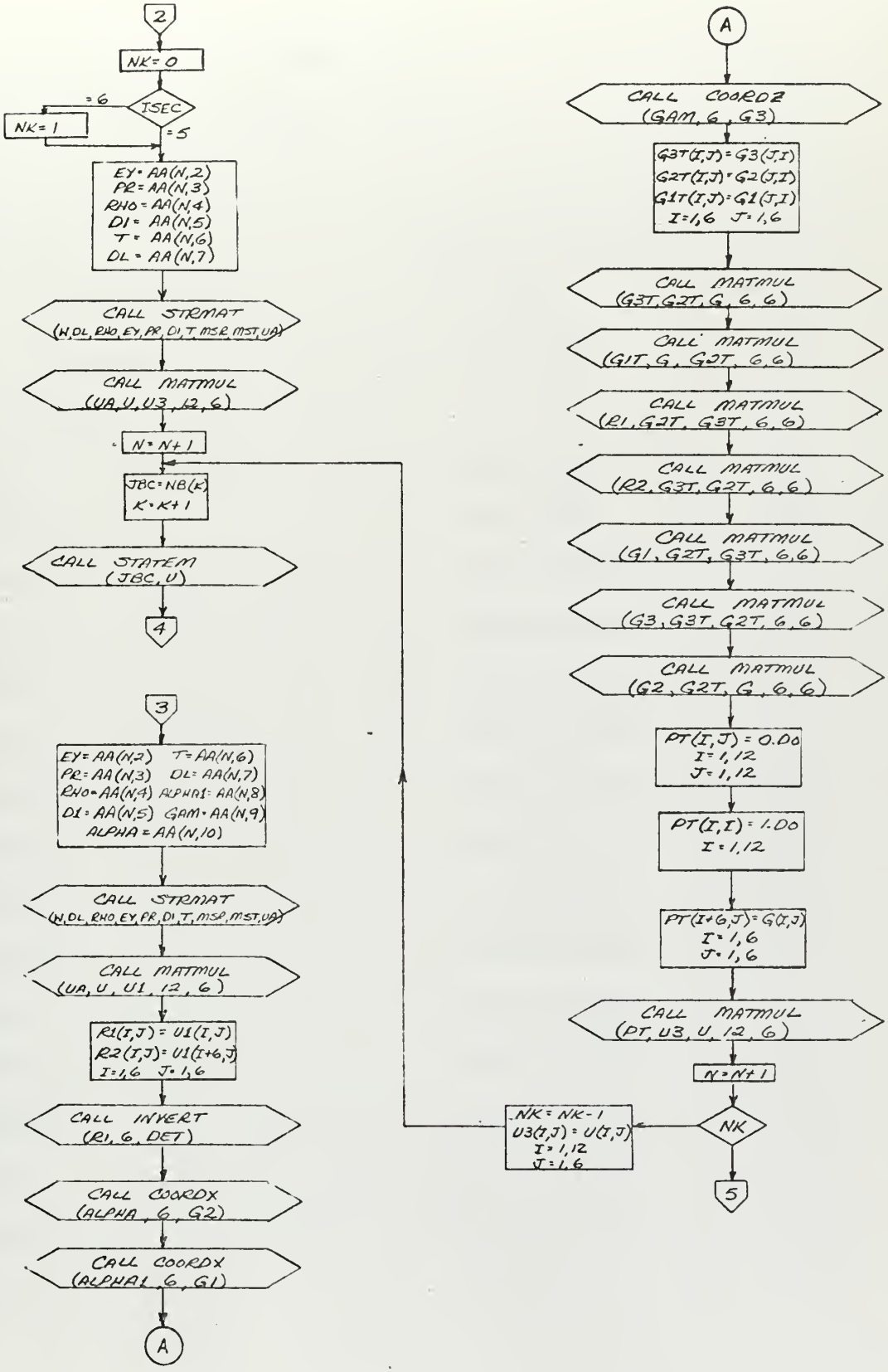

BRANCH END POINT (JUNCTION WITH MAIN MEMBER)
12
DL = DSQRT (A(1)**2 + A(2) **2 + A(3)**2)
AA(N-1,1)
= 2.
DL = DL - BL
DL
TO OUTPUT
AA(N,1) = Z. AA(N,5) = DI
AA(N,2) = EY AA(N,6) = T
AA(N,3) = PR AA(N,7) = DL
AA(N,4) = RHO
B(1) = PXB - PX(2)
B(2) = PYB - PY(2)
B(3) = PZB - PZ(2)
CALL UVEC (A,B,UXP,UYP,UZP,GAM)
CALL ANGLE (UY,UYP, UX, AL1)
AA(N,8) = AL1
CALL ANGLE (UXP, SX, UZP, GAM1)
AA(N,9) = GAM
GAM1 = DACOS(GAM)
UZDP(I) = UZP(I)
UXDP(I) = UXP(I) * DCOS(GAM1) - UYP(I) * DSIN(GAM1)
UYDP(I) = UYP(I) * DSIN(GAM1) + UYP(I) * DCOS(GAM1)
I = 1, 3
CALL ANGLE (UYDP, SY, SX, AL)
AA(N,10) = AL
N = N + 1
KB = KB + 1
NIB = 0
KB
= 1
17
= 2
UX(I) = SX(I)
UY(I) = SY(I)
UZ(I) = SZ(I)
I = 1, 3
PX(1) = PX(2) YR = QR
PY(1) = PY(2) YEY = QEY
PZ(1) = PZ(2) YPR = QPR
YX = PXB YD = QD
YY = PYB YT = QT
YZ = PZB YRHO = QRHO
NY = NQ
16


ITERATION
EYT = 0. D1T = 0.
PRT = 0 TT = 0.
RHOT = 0. DLT = 0.
NN = 0
NN = NN + 1
NSEC = AA(NN,1)
NSEC
= 2
DL = AA(NN,7)
DL = AA(NN,7) * AA(NN,9)
DLT = DLT + DL
EYT = EYT + AA(NN,2) * DL
PRT = PRT + AA(NN,3) * DL
RHOT = RHOT + AA(NN,4) * DL
D1T = D1T + AA(NN,5) * DL
TT = TT + AA(NN,6) * DL
NSEC
= 1,2,3
= 5
= 6
= 4
NN = NN + 1
NSEC = AA(NN,1)
≠ 7
= 7
DEN = DLT
EY = EYT/DEN
PR = PRT/DEN
RHO = RHOT/DEN
D1 = D1T/DEN
T = TT/DEN
DL = DLT
W1 = .100
CALL WSTART
(W1, DL, RHO, EY, PR, D1, T, MSR, DW2
W2 = W1 + 5.00
CALL WSTART
W1, DL, RHO, EY, PR, D1, T, MSR, DW2
NDISCR =
DW1/DABS(DW1)
+DW2/DABS(DW2)
NDISCR
≠ 0
= 0
W1 = W2
DW1 = DW2
X1 = (DABS(DW1)
*(W2-W1)) /
(DABS(DW1) +
DABS(DW2))
18


18
WINCR = (W1 + X1)/4.
FINCR = WINCR
NM = 1
W1 = .0100
DW1 = FRDET(W1)
X(1) = W1
Y(1) = DW1
YMAX = DW1
YMIN = DW1
XMIN = 0.00
M = 2
W2 = W1 + WINCR
XMAX = W2
DW2 = FRDET(W2)
DABS(DW2)
>10^60 or < 10^-60
WRITE: PROBLEM TERMINATED
DW2
< YMIN
YMIN = DW2
> YMAX
YMAX = DW2
Y(M) = W2
Y(M) = DW2
NDISCR = DW1/DABS(DW1) + DW2/DABS(DW2)
M = M + 1
NDISCR
W1 = W2
DW1 = DW2
EPS = FINCR/10.
CALL CROOT (WN, DWN, FRDET, W1, W2, EPS, NIT, IER)
IER
= 1
WRITE: ERROR
= 2
WRITE: ERROR
FN(NM) = WN
WRITE: NM, FN(NM), DWN
NM = NM + 1
NM - NM1
≤ 0
> 0
NM
≤ 2
> 2
WINCR = FN(NM-1) - FN(NM-2)/10.00
W1 = WN + 1.00
DW1 = FRDET(W1)
X(M) = WN
Y(M) = DWN
M = M + 1
X(M) = WN
Y(M) = DWN
CALL SORT (X, Y, M)
CALL FRPLOT (X, Y, M, XMIN, XMAX, YMIN, YMAX, 0)
STOP


START
DIMENSION
MSR = MS(1)
MST = MS(2)
LBC = NB(1)
N = 1
K = 3
CALL STATEM (LBC, U)
4
5
ISEC = AA(N,1)
ISEC
= 2
EY = AA(N,2) T = AA(N,6)
PR = AA(N,3) R = AA(N,7)
RHO = AA(N,4) GAM = AA(N,9)
DI = AA(N,5) NSS = AA(N,10)
ALPHA = AA(N,8)
ALPHA
≤ .01
UI(I,J) = U(I,J) I = 1,12 J = 1,6
> .01
CALL COORDX (ALPHA, 12, CTX12)
CALL MATMUL (CTX12, U, UI, 12, 6)
GAMS = GAM (FLOAT(NSS))
CALL POINT (RHO, DI, T, GAMS, R, H, UA)
CALL CURMAT (EY, PR, DI, R, T, GAMS, MSR, UB)
CALL MATMUL (UB, U2, UI, 12, 6)
CALL MATMUL (UB, U2, UI, 12, 6)
NSS = NSS-1
NSS
> 0
U(I,J) = UI(I,J) I = 1,12 J = 1,6
N = N+1
= 4
1
= 3
EY = AA(N,2) T = AA(N,6)
PR = AA(N,3) DL = AA(N,7)
RHO = AA(N,4) ALPHA = AA(N,8)
DI = AA(N,5) GAM = AA(N,9)
ALPHA
≤ .01
UI(I,J) = U(I,J) I = 1,12 J = 1,6
> .01
CALL COORDX (ALPHA, 12, CTX12)
CALL MATMUL (CTX12, U, UI, 12, 6)
CALL STRMAT (W, DL, RHO, EY, PR, DI, T, MSR, MST, UA)
CALL MATMUL (UA, U1, U2, 12, 6)
CALL COORDZ (GAM, 12, CT212)
CALL MATMUL (CT212, U2, U, 12, 6)
N = N+1
= 5,6
2
= 7
3
= 1
EY = AA(N,2)
PR = AA(N,3)
RHO = AA(N,4)
DI = AA(N,5)
T = AA(N,6)
DL = AA(N,7)
CALL STRMAT (W, DL, RHO, EY, PR, DI, T, MSR, MST, UA)
CALL MATMUL (UA, U, U1, 12, 6)
U(I,J) = UI(I,J) I = 1,12 J = 1,6
N = N+1


1
EY = AA(N,2)
PR = AA(N,3)
RHO = AA(N,4)
D1 = AA(N,5)
T = AA(N,6)
DL = AA(N,7)
DL
= 0.
> 0.
U1(I,J) = U(I,J)
I = 1,6
J = 1,6
CALL STRMAT
(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)
CALL MATMUL
(UA,U,U1,12,6)
MBC = NB(2)
MBC
= 0
= 1
= 2
DD(I,J) = U1(I+6,J)
I = 1,6
J = 1,6
DD(I,J) = U1(I,J)
I = 1,3 J = 1,6
DD(I,J) = U1(I+6,J)
I = 4,6 J = 1,6
DD(I,J) = U1(I,J)
I = 1,6
J = 1,6
CALL DETER
(6,DD,D)
FRDET = D
RETURN


2
NK=0
ISEC
=6
NK=1
=5
EY=AA(N,2)
PR=AA(N,3)
RHO=AA(N,4)
DI=AA(N,5)
T=AA(N,6)
DL=AA(N,7)
CALL STRMAT
(N,DL,RHO,EY,PR,DI,T,MSR,MST,UA)
CALL MATMUL
(UA,U,U3,12,6)
N=N+1
JBC=NB(K)
K=K+1
CALL STATEM
(JBC,U)
4
3
EY=AA(N,2) T=AA(N,6)
PR=AA(N,3) DL=AA(N,7)
RHO=AA(N,4) ALPHA1=AA(N,8)
DI=AA(N,5) GAM=AA(N,9)
ALPHA=AA(N,10)
CALL STRMAT
(N,DL,RHO,EY,PR,DI,T,MSR,MST,UA)
CALL MATMUL
(UA,U,U1,12,6)
R1(I,J)=U1(I,J)
R2(I,J)=U1(I+6,J)
I=1,6 J=1,6
CALL INVERT
(R1,6,DET)
CALL COORDX
(ALPHA,6,G2)
CALL COORDX
(ALPHA1,6,G1)
A
A
CALL COORD2
(GAM,6,G3)
G3T(I,J)=G3(J,I)
G2T(I,J)=G2(J,I)
G1T(I,J)=G1(J,I)
I=1,6 J=1,6
CALL MATMUL
(G3T,G2T,G,6,6)
CALL MATMUL
(G1T,G,G2T,6,6)
CALL MATMUL
(R1,G2T,G3T,6,6)
CALL MATMUL
(R2,G3T,G2T,6,6)
CALL MATMUL
(G1,G2T,G3T,6,6)
CALL MATMUL
(G3,G3T,G2T,6,6)
CALL MATMUL
(G2,G2T,G,6,6)
PT(I,J)=0.00
I=1,12
J=1,12
PT(I,I)=1.00
I=1,12
PT(I+6,J)=G(I,J)
I=1,6
J=1,6
CALL MATMUL
(PT,U3,U,12,6)
N=N+1
NK
NK=NK-1
U3(I,J)=U(I,J)
I=1,12
J=1,6
5

APPENDIX E

PROGRAM LISTING

1. Index

```
C*****************************************************          00000010
C*********                                           **********  00000020
C*********              INSTRUCTIONS FOR             **********  00000030
C*********        THE USE OF PROGRAM VIBREL          **********  00000040
C*********                                           **********  00000050
C*****************************************************          00000060
C                                                               00000070
C                                                               00000080
C                                                               00000090
C REFERENCE: RUDOLF,C.D., "3-D PIPING VIBRATIONS VIA TRANSFER   00000100
C        MATRICES" , U.S. NAVAL POSTGRADUATE SCHOOL THESIS, DEC,1971.  00000110
C                                                               00000120
C                                                               00000130
C A.GENERAL REMARKS                                             00000140
C                                                               00000150
C     PROGRAM VIBREL MAY BE USED IN THE ANALYSIS OF A           00000160
C   THREE DIMENSIONAL PIPING SYSTEM HAVING ARBITRARY CON-       00000170
C   FIGURATION. THE CAPABILITIES AND LIMITATIONS ARE:           00000180
C                                                               00000190
C        1.THE PIPING SYSTEM MUST BE COMPOSED OF A MAIN MEMBER AND  00000200
C          ANY NUMBER OF SIMPLE BRANCHES. THE MAIN MEMBER IS DESIG-  00000210
C          NATED AS THE LONGEST CONTINUOUS RUN OF PIPE FROM WHICH ALL  00000220
C          BRANCHES EMANATE. A MAXIMUM OF TWO BRANCHES CAN JOIN THE  00000230
C          MAIN MEMBER AT ANY ONE POINT AND THERE CAN BE NO BRANCHES  00000240
C          EMANATING FROM OTHER BRANCHES. THE SYSTEM CANNOT CONTAIN  00000250
C          LOOPS.                                               00000260
C                                                               00000270
C        2.THERE MAY BE NO CURVATURE OF THE PIPING OR ABRUPT CHANGE  00000280
C          IN DIRECTION OF THE MAIN MEMBER AT A BRANCH JUNCTION  00000290
C          POINT.                                               00000300
C                                                               00000310
C        3.PIPING HANGERS CANNOT BE HANDLED DIRECTLY BY PROGRAM  00000320
C          VIBREL. A SPRING HANGER MAY BE APPROXIMATED AS A VERY  00000330
C          SMALL PIPE OF A HIGH MODULUS MATERIAL , INCLUDED IN  00000340
C          THE SYSTEM AS A BRANCH EMANATING FROM THE MAIN MEMBER.  00000350
C          RELATIVELY SMALL PIPING ACCESSORIES SUCH AS VALVES CAN BE  00000360
C          APPROXIMATED BY VARYING THE  PROPERTIES OF A SMALL STRAIGHT  00000370
C          SECTION OF PIPE.                                     00000380
C                                                               00000390
C                                                               00000400
C        4.PIPE CROSS SECTION AND PROPERTIES MAY VARY FROM SECTION TO  00000410
C          SECTION BUT MAY NOT VARY WITHIN A SECTION.           00000420
C                                                               00000430
C        5.SHEAR DEFLECTION AND ROTARY INERTIA MAY BE INCLUDED IN  00000440
C          THE ANALYSIS AT THE DISCRETION OF THE USER.          00000450
C                                                               00000460
C        6.THE PROGRAM IS CAPABLE OF ANALYZING ONE OR MORE SYSTEMS  00000470
C          DURING A SINGLE COMPUTER RUN.                        00000480
```

```
C                                                                       00000490
C                                                                       00000500
C 8.COORDINATE SYSTEM AND WORKING POINTS                                00000510
C                                                                       00000520
C    THE PIPING SYSTEM GEOMETRY IS DETERMINED BY WORKING POINTS,        00000530
C   THE COORDINATES OF WHICH ARE MEASURED FROM THE ORIGIN OF A SET OF    00000540
C   MUTUALLY ORTHOGONAL X,Y, AND Z AXES. THE ORIGIN IS LOCATED          00000550
C   ARBITRARILY AND REMAINS FIXED IN SPACE. ALL WORKING POINT COOR-     00000560
C   DINATES ARE MEASURED WITH THE PIPING SYSTEM IN ITS QUIESCENT        00000570
C   CONFIGURATION.                                                      00000580
C    THERE IS PRESENTLY NO  PROVISION FOR ASSOCIATING AN IDENTFYING     00000590
C   NUMBER OR SYMBOL WITH ANY POINTS OF PARTICULAR SIGNIFICANCE         00000600
C   IN THE SYSTEM.                                                      00000610
C    HOWEVER, THE INPUT EMPLOYS EIGHT TYPES OF WORKING POINTS AS        00000620
C   INDICATED IN THE FOLLOWING TABLE. A NINTH TYPE IS REQUIRED          00000630
C   FOR INTERNAL PROGRAM PURPOSES IN THE CASE OF DUAL BRANCHES,I.E.,    00000640
C   TWO BRANCHES EMANATING FROM THE SAME POINT OF THE MAIN MEMBER.      00000650
C                                                                       00000660
C        POINT TYPE CODE                  DEFINITION                    00000670
C                                                                       00000680
C              1        STARTING POINT(MAIN)- FIRST POINT OF THE SYSTEM 00000690
C                       MAIN MEMBER, HENCEFORTH REFERRED TO AS THE LEFT 00000700
C                       END. IT DETERMINES A SYSTEM BOUNDARY.           00000710
C                                                                       00000720
C              2        STARTING POINT(BRANCH)- FIRST POINT OF A BRANCH 00000730
C                       FARTHEST FROM THE INTERSECTION WITH THE MAIN    00000740
C                       MEMBER. IT ALSO DETERMINES A SYSTEM BOUNDARY.   00000750
C                                                                       00000760
C                                                                       00000770
C              3        PROPERTY VARIATION POINT-A POINT OCCURRING IN   00000780
C                       A STRAIGHT RUN OF PIPE WHERE EITHER  CROSS      00000790
C                       SECTION OR PROPERTIES CHANGE. CROSS SECTION     00000800
C                       OR PROPERTIES MAY ALSO CHANGE AT ANY OTHER      00000810
C                       WORKING POINT.                                  00000820
C                                                                       00000830
C              4        BEND POINT-THE POINT AT WHICH THE TANGENTS TO   00000840
C                       THE TWO ENDS OF A CURVED SECTION OF PIPE        00000850
C                       INTERSECT. THE BEND POINT AND RADIUS OF CURV-   00000860
C                       ATURE GEOMETRICALLY DESCRIBE  A CURVED SECTION  00000870
C                       OF PIPE.                                        00000880
C                                                                       00000890
C              5        CORNER POINT-THE POINT AT WHICH A SECTION OF    00000900
C                       PIPE CHANGES DIRECTION WITH NO APPRECIABLE      00000910
C                       RADIUS OF CURVATURE.                            00000920
C                                                                       00000930
C              6        BRANCH POINT-THE POINT IN THE MAIN MEMBER FROM  00000940
C                       WHICH A BRANCH EMANATES.                        00000950
C                                                                       00000960
```

```
C                7       DUAL-BRANCH POINT-THE POINT IN THE MAIN MEMBER  00000970
C                        FROM WHICH TWO BRANCHES EMANATE.                00000980
C                                                                        00000990
C                8       END POINT-EITHER THE FINAL POINT OF THE MAIN    00001000
C                        MEMBER, HENCEFORTH CALLED THE RIGHT END, OR THE 00001010
C                        POINT WHERE A BRANCH JOINS THE MAIN MEMBER. THE 00001020
C                        BRANCH END POINT COORDINATES ARE COINCIDENTAL   00001030
C                        WITH THE CORRESPONDING BRANCH POINT(6) OR DUAL- 00001040
C                        BRANCH POINT(7).                                00001050
C                                                                        00001060
C                9       DUAL-BRANCH REPEAT- THE MAIN MEMBER WORKING     00001070
C                        POINT IMMEDIATELY FOLLOWING A DUAL-BRANCH POINT.00001080
C                        IT IS REPEATED IN THE CARD INPUT SEQUENCE       00001090
C                        TO CAUSE PROPER INTERNAL PROGRAM OPERATION.     00001100
C                                                                        00001110
C                                                                        00001120
C  C.SEQUENCE OF POINTS FOR PROGRAM DATA INPUT                           00001130
C                                                                        00001140
C     AFTER THE STARTING POINT OF THE MAIN MEMBER(1), WORKING POINTS     00001150
C   ARE LISTED IN SEQUENCE WHILE TRAVERSING FROM THE LEFT END TOWARD     00001160
C   THE RIGHT END UNTIL A BRANCH OR DUAL-BRANCH POINT IS ENCOUNTERED.    00001170
C   AFTER A (SINGLE) BRANCH POINT(6) IS LISTED,THE SEQUENCE SKIPS TO THE00001180
C   BRANCH STARTING POINT(2) AND CONTINUES THROUGH THE BRANCH END POINT 00001190
C   (8) BACK ONTO THE MAIN MEMBER AND PROCEEDS AGAIN TOWARD THE RIGHT   00001200
C   END. THE BRANCH POINT(6) AND BRANCH END POINT(8) HAVE IDENTICAL      00001210
C   COORDINATES.                                                         00001220
C                                                                        00001230
C     THE SEQUENCE IS THE SAME FOR A DUAL-BRANCH EXCEPT THAT AFTER       00001240
C   THE FIRST BRANCH END POINT(8),THE DUAL-BRANCH REPEAT(9), HAVING      00001250
C   THE COORDINATES OF THE FOLLOWING MAIN MEMBER WORKING POINT, IS       00001260
C   LISTED. THEN THE SEQUENCE PROCEEDS WITH THE STARTING POINT OF THE    00001270
C   SECOND BRANCH AND CONTINUES AS IN THE CASE OF A SINGLE BRANCH. THE   00001280
C   ORDER OF LISTING THE TWO BRANCHES IS INCONSEQUENTIAL.                00001290
C                                                                        00001300
C     SINCE PROGRAM VIBREL AUTOMATICALLY DEALS WITH THE GEOMETRY         00001310
C   OF BENDS, PROPERLY LOCATING THE POINTS OF TANGENCY, IT IS NOT        00001320
C   NECESSARY TO LIST WORKING POINTS FOR THE ENDS OF A CURVED SECTION    00001330
C   OF PIPE UNLESS PROPERTIES OR CROSS SECTION CHANGE AT THE             00001340
C   END POINTS OR SOMEWHERE WITHIN THE BEND ITSELF. FOR FURTHER          00001350
C   CLARIFICATION SEE APPENDIX C OF THE REFERENCE THESIS.                00001360
C                                                                        00001370
C                                                                        00001380
C  D.PIPING PROPERTIES AND DATA READ-IN FORMAT                           00001390
C                                                                        00001400
C     THE PROPERTIES ASSOCIATED WITH EACH WORKING POINT ARE THOSE OF     00001410
C   THE SECTION OF PIPE BETWEEN IT AND THE PRECEDING WORKING POINT.      00001420
C   IF MODULUS OF ELASTICITY, POISSON'S RATIO, OUTSIDE DIAMETER, WALL    00001430
C   THICKNESS, OR WEIGHT DENSITY ARE REPEATED FROM WORKING POINT TO WOR-00001440
```

```
C   KING POINT, A BLANK OR A ZERO MAY BE SUBSTITUTED IN THE                  00001450
C   APPROPRIATE COLUMN FOR THE REPEATED VALUES UNTIL A CHANGE IS             00001460
C   ENCOUNTERED. FOR WORKING POINTS OTHER THAN BEND POINTS THE RADIUS        00001470
C   OF CURVATURE IS CONSIDERED TO BE ZERO FOR PROGRAM PURPOSES,              00001480
C   ALTHOUGH FOR STRAIGHT PIPE IT WOULD BE INFINITE.                         00001490
C                                                                            00001500
C   DATA FORMAT:                                                             00001510
C                                                                            00001520
C      1)CARD A  - USED ONLY ONCE                                            00001530
C                                                                            00001540
C         COL 1-2      NUMBER OF PIPING SYSTEMS TO BE            I2        00001550
C                        ANALYZED                                            00001560
C                                                                            00001570
C      2)CARD B  - USED ONCE FOR EACH SYSTEM ANALYZED                        00001580
C                                                                            00001590
C         COL 1        SHEAR DEFLECTION/ROTARY INERTIA           I1        00001600
C                        DISCRIMINANT                                        00001610
C                        1- S.D./R.I. CONSIDERED                             00001620
C                        0- S.D./R.I. NOT CONSIDERED                         00001630
C                                                                            00001640
C         COL 2        STRAIGHT SECTION TEST DISCRIMINANT        I1        00001650
C                        1- SYSTEM COMPOSED OF ONLY ONE                      00001660
C                           STRAIGHT SECTION OF PIPE                         00001670
C                        0- ANY OTHER SYSTEM                                 00001680
C                                                                            00001690
C         COL 3-4      CURVED SUBSECTION OVERRIDE                I2        00001700
C                        IF ZERO, PROGRAM DETERMINES                         00001710
C                        NUMBER OF SUBSECTIONS FOR CURVED                    00001720
C                        PIPE. ANY NUMBER OTHER THAN                         00001730
C                        ZERO PRESCRIBES THAT MANY SUB-                      00001740
C                        SECTIONS FOR EACH BEND CALCULATION                  00001750
C                                                                            00001760
C         COL 5-6      NUMBER OF MODE FREQUENCIES DESIRED        I2        00001770
C                        IN THE ANALYSIS                                     00001780
C                                                                            00001790
C      3)SUCCEEDING CARDS- ONE FOR EACH WORKING POINT                        00001800
C                                                                            00001810
C          A)STARTING POINT(MAIN)                                            00001820
C                                                                            00001830
C               COL 1      POINT TYPE CODE                     I1        00001840
C                          (SEE SECTION B)                                   00001850
C                                                                            00001860
C               COL 2-7    X COORDINATE, INCHES                F6.1      00001870
C                                                                            00001880
C               COL 8-13   Y COORDINATE, INCHES                F6.1      00001890
C                                                                            00001900
C               COL 14-19  Z COORDINATE, INCHES                F6.1      00001910
C                                                                            00001920
```

```
C                COL 20    LEFT END BOUNDARY CONDITION CODE        I1      00001930
C                            B.C. CODES:  0- FREE                         00001940
C                                         1- BUILT-IN                     00001950
C                                         2- BALL JOINT                   00001960
C                                                                         00001970
C                COL 21    RIGHT END BOUNDARY CONDITION CODE       I1      00001980
C                                                                         00001990
C           B)STARTING POINT(BRANCH)                                      00002000
C                                                                         00002010
C                COLUMNS 1-20 SAME AS FOR MAIN MEMBER                     00002020
C                STARTING POINT.(RIGHT END B.C. CODE IS                   00002030
C                OMITTED)                                                 00002040
C                                                                         00002050
C           C)ALL OTHER WORKING POINTS                                    00002060
C                                                                         00002070
C                COL 1     POINT TYPE CODE(SEE SECTION B)        I1      00002080
C                                                                         00002090
C                COL 2-7   X COORDINATE, INCHES                  F6.1    00002100
C                                                                         00002110
C                COL 8-13  Y COORDINATE,  INCHES                 F6.1    00002120
C                                                                         00002130
C                COL 14-19 Z COORDINATE, INCHES                  F6.1    00002140
C                                                                         00002150
C                COL 20-24 RADIUS OF CURVATURE,  INCHES          F5.1    00002160
C                            0.0 OR BLANK FOR OTHER                       00002170
C                            THAN BEND POINTS.                            00002180
C                                                                         00002190
C                COL 25-32 MODULUS OF ELASTICITY,  PSI           E8.1    00002200
C                                                                         00002210
C                COL 33-35 POISSON'S RATIO                       F3.2    00002220
C                                                                         00002230
C                COL 36-40 OUTSIDE PIPE DIAMETER, INCHES         F5.2    00002240
C                                                                         00002250
C                COL 41-46 WALL THICKNESS, INCHES                F6.4    00002260
C                                                                         00002270
C                COL 47-52 WEIGHT DENSITY, LB/CUBIC FT           F6.1    00002280
C                                                                         00002290
C           A DUAL BRANCH REPEAT(9) CARD NEED ONLY CONTAIN THE INFORMATION 00002300
C           REQUIRED IN COLUMNS 1-19.                                     00002310
C                                                                         00002320
C           EACH SYSTEM ANALYZED MUST CONTAIN ONE CARD B  AND THE APPROPRIATE00002330
C           WORKING POINT CARDS. THE DATA DECKS (AS MANY AS WERE LISTED ON CARD 00002340
C           A ) ARE PLACED BACK-TO-BACK BEHIND THE CARD A. FOR EACH SYSTEM THE 00002350
C           FINAL TYPE 8 CARD SIGNALS THE START OF THE COMPUTATION FOR THAT 00002360
C           PARTICULAR SYSTEM.                                            00002370
C                                                                         00002380
C                                                                         00002390
C                                                                         00002400
C      E. PROGRAM OUTPUT
```

```
C                                                                       00002410
C     IN ADDITION TO PRINTING THE INPUT DATA AS READ FROM THE DATA      00002420
C   CARDS, PROGRAM VIBREL PRINTS THE PROPERTIES AND GEOMETRY COMPUTED   00002430
C   FOR EACH SECTION OF PIPE. INCLUDED IN THIS OUTPUT ARE THE FOLLOWING 00002440
C   SECTION IDENTIFICATION CODES:                                       00002450
C                                                                       00002460
C              1- STRAIGHT SECTION                                      00002470
C                                                                       00002480
C              2- CURVED SECTION                                        00002490
C                                                                       00002500
C              3- STRAIGHT SECTION IMMEDIATELY PRECEEDING A CORNER      00002510
C                                                                       00002520
C              4- LAST SECTION OF THE MAIN MEMBER                       00002530
C                                                                       00002540
C              5- MAIN MEMBER SECTION IMMEDIATELY PRECEEDING A BRANCH   00002550
C                                                                       00002560
C              6- MAIN MEMBER SECTION IMMEDIATELY PRECEEDING A DUAL-    00002570
C                  BRANCH                                               00002580
C                                                                       00002590
C              7- BRANCH SECTION WHICH JOINS THE MAIN MEMBER            00002600
C                                                                       00002610
C     THE LAST THREE COLUMNS OF PROPERTIES AND GEOMETRY OUTPUT CON-     00002620
C   TAIN APPROPRIATE ROTATION ANGLES AND CURVED PIPE SUBSECTIONING      00002630
C   INFORMATION. THE FIRST CONTAINS ALPHA1, A LOCAL X-AXIS COORDINATE   00002640
C   TRANSFORMATION ANGLE. THE SECOND COLUMN LISTS THE VALUE OF GAMMA,    00002650
C   A LOCAL Z-AXIS COORDINATE TRANSFORMATION ANGLE. THE FINAL COLUMN    00002660
C   CONTAINS EITHER THE FINAL X-AXIS COORDINATE TRANSFORMATION ANGLE    00002670
C   FOR A BRANCH JOINING THE MAIN MEMBER, OR FOR CURVED SECTIONS IT     00002680
C   LISTS THE NUMBER OF SUBSECTIONS USED FOR COMPUTATIONAL PURPOSES.    00002690
C                                                                       00002700
C     FOLLOWING THE PROPERTIES AND GEOMETRY OUTPUT THE BOUNDARY CON-    00002710
C   DITIONS AND THE NATURAL FREQUENCIES ARE PRINTED. A GRAPH OF THE     00002720
C   VALUE OF FREQUENCY DETERMINANT VERSUS FREQUENCY IS                  00002730
C   ALSO DISPLAYED AS AN AID IN DETECTING IF THE PROGRAM HAS SKIPPED    00002740
C   ANY NATURAL FREQUENCIES AS A RESULT OF THE FREQUENCY INCREMENT      00002750
C   BEING TOO LARGE.                                                    00002760
C                                                                       00002770
C                                                                       00002780
C                                                                       00002790
C                                                                       00002800
C                                                                       00002810
C*************************************************************        00002820
C********                                                    *        00002830
C********          PROGRAM    VIBREL                       *        00002840
C********                                                    *        00002850
C*************************************************************        00002860
      IMPLICIT REAL*8(A-H,O-Z)                                          00002870
      DIMENSION PX(3),PY(3),PZ(3),A(3),B(3),UX(3),UY(3),UZ(3),UXP(3),UYP00002880
```

```
     1(3),UZP(3),SX(3),SY(3),SZ(3),UXDP(3),UYDP(3),UZDP(3),X(400),Y(400)00002890
     2,FN(30)                                                           00002900
      COMMON/VIB1/AA(100,10)                                            00002910
      COMMON/VIB2/NB(50)                                                00002920
      COMMON/VIB3/MS(2)                                                 00002930
      EXTERNAL FRDET                                                    00002940
      NPROB1=1                                                          00002950
C                                                                       00002960
CC   READ NUMBER OF DIFFERENT SYSTEMS TO BE ANALYZED                    00002970
C                                                                       00002980
      READ(5,45)NPROB                                                   00002990
C                                                                       00003000
CC   READ SHEAR DEFLECTION/ROTARY INERTIA DISCRIMINANT, SINGLE STRAIGHT 00003010
CC     SECTION DISCRIMINANT,CURVED SUBSECTION OVERRIDE,AND NUMBER       00003020
CC     OF MODES SOUGHT                                                  00003030
C                                                                       00003040
   23 READ(5,50)MSR,MST,MCS,NMO                                         00003050
      DO 30 I=1,100                                                     00003060
      DO 30 J=1,10                                                      00003070
   30 AA(I,J)=0.0D0                                                     00003080
      DO 40 I=1,10                                                      00003090
   40 NB(I)=0                                                           00003100
      MS(1)=MSR                                                         00003110
      MS(2)=MST                                                         00003120
C                                                                       00003130
CC   READ STARTING POINT COORDINATES                                    00003140
C                                                                       00003150
      READ(5,60)NID,PX(1),PY(1),PZ(1),LBC,MBC                           00003160
C                                                                       00003170
CC   WRITE TOP PORTION OF OUTPUT FORMAT AND INPUT DATA                  00003180
C                                                                       00003190
      WRITE(6,70)                                                       00003200
   45 FORMAT(I2)                                                        00003210
   50 FORMAT(2I1,2I2)                                                   00003220
   60 FORMAT(I1,3F6.1,2I1)                                              00003230
   61 FORMAT(I1,3F6.1,I1)                                               00003240
   62 FORMAT(I1,3F6.1,F5.1,E8.1,F3.2,F5.2,F6.4,F6.1)                    00003250
   70 FORMAT('1',5X,'**************************************************00003260
     1**********************',/,6X,'*',71X,'*',/,6X,'*',15X,'PROGRAM  VI00003270
     2BREL',21X,'C.D.RUDOLF',10X,'*',/,6X,'*',71X,'*',/,6X,'************00003280
     3**************************************************',//)00003290
   76 FORMAT(6X,'INPUT DATA:')                                          00003300
   77 FORMAT(/,6X,'PROGRAM VIBREL COMPUTES THE NATURAL FREQUENCIES OF VI00003310
     1BRATION',/,9X,'OF ANY RANDOMLY ARRANGED 3-DIMENSIONAL PIPING SYSTE00003320
     2M',/,6X,'PIPING HANGERS AND PROJECTIVE COMPLEXITIES ARE NOT INCLUD00003330
     3ED')                                                              00003340
   78 FORMAT(///,14X,'X',6X,'Y',6X,'Z',5X,'RAD OF',2X,'ELASTIC',3X,'POIS00003350
     1SONS',10X,'WALL',14X,'LEFT BC',4X,'RIGHT BC',/,7X,'ID',3X,'COORD',00003360
```

```
     22X,'COORD',2X,'COORD',4X,'CURV',3X,'MODULUS',4X,'RATIO',5X,'O.D.',00003370
     32X,'THICK',3X,' WT/VOL ',4X,'CODE',8X,'CODE',/)                    00003380
  81 FORMAT(7X,I1,3X,F6.1,1X,F6.1,1X,F6.1,59X,I1,11X,I1)                 00003390
  82 FORMAT(7X,I1,3X,F6.1,1X,F6.1,1X,F6.1,59X,I1)                        00003400
  83 FORMAT(7X,I1,3X,F6.1,1X,F6.1,1X,F6.1,3X,F5.1,2X,E9.3,4X,F3.2,5X,F500003410
    1.2,2X,F6.4,2X,F6.1)                                                00003420
  85 FORMAT(//,6X,'**********************************************************0003430
    1**********',//)                                                    00003440
     WRITE(6,77)                                                        00003450
     WRITE(6,85)                                                        00003460
     WRITE(6,84)NPROB1                                                  00003470
     WRITE(6,85)                                                        00003480
  84 FORMAT(6X,'PROBLEM #',I2)                                          00003490
     IF(MSR.EQ.0)GO TO 87                                               00003500
     WRITE(6,86)                                                        00003510
  86 FORMAT(6X,'THE EFFECTS OF SHEAR DEFLECTION AND ROTARY INERTIA ARE 00003520
    1CONSIDERED IN THIS PROBLEM')                                       00003530
     GO TO 89                                                           00003540
  87 WRITE(6,88)                                                        00003550
  88 FORMAT(6X,'THE EFFECTS OF SHEAR DEFLECTION AND ROTARY INERTIA ARE 00003560
    1NOT CONSIDERED IN THIS PROBLEM')                                   00003570
  89 WRITE(6,85)                                                        00003580
     WRITE(6,76)                                                        00003590
     WRITE(6,78)                                                        00003600
     WRITE(6,81)NID,PX(1),PY(1),PZ(1),LBC,MBC                           00003610
     NIV=0                                                              00003620
C                                                                       00003630
C    SET MAXIMUM NUMBER OF ITERATIONS ALLOWABLE FOR BISECTION WHEN      00003640
C      CALCULATING THE NATURAL FREQUENCY IN SUBROUTINE DROOT            00003650
C                                                                       00003660
     NIT=15                                                             00003670
     NIB=0                                                              00003680
     PI=3.141592653589793                                               00003690
     NB(1)=LBC                                                          00003700
     NB(2)=MBC                                                          00003710
     K=2                                                                00003720
     N=1                                                                00003730
C                                                                       00003740
C    READ NEXT POINT AND SECTION PROPERTIES                             00003750
C                                                                       00003760
 100 READ(5,62)NV,VX,VY,VZ,VR,VEY,VPR,VD,VT,VRHO                        00003770
     WRITE(6,83)NV,VX,VY,VZ,VR,VEY,VPR,VD,VT,VRHO                       00003780
 105 IF((NV.LT.4).OR.(NV.GT.5))GO TO 110                                00003790
C                                                                       00003800
C      IF BEND OR CORNER POINT READ THE FOLLOWING POINT                 00003810
C                                                                       00003820
     READ(5,62)NQ,PXB,PYB,PZB,QR,QEY,QPR,QD,QT,QRHO                     00003830
     WRITE(6,83)NQ,PXB,PYB,PZB,QR,QEY,QPR,QD,QT,QRHO                    00003840
```

```
  110 NID=NV                                                                00003850
      PX(2)=VX                                                              00003860
      PY(2)=VY                                                              00003870
      PZ(2)=VZ                                                              00003880
      R=VR                                                                  00003890
C                                                                           00003900
C    IF THE VALUES OF EY,PR,D1,T, OR RHO ARE ZERO, RETAIN THE               00003910
C      PREVIOUS VALUE                                                       00003920
C                                                                           00003930
      IF(VEY.GT.1.D-9)EY=VEY                                                00003940
      IF(VPR.GT.1.D-9)PR=VPR                                                00003950
      IF(VD.GT.1.D-9)D1=VD                                                  00003960
      IF(VT.GT.1.D-9)T=VT                                                   00003970
      IF(VRHO.GT.1.D-9)RHO=VRHO                                             00003980
      A(1)=PX(2)-PX(1)                                                      00003990
      A(2)=PY(2)-PY(1)                                                      00004000
      A(3)=PZ(2)-PZ(1)                                                      00004010
      IF(NIV.GT.0)GO TO 140                                                 00004020
      IF((NID.EQ.4).OR.(NID.EQ.5))GO TO 136                                 00004030
C                                                                           00004040
C////////////////////////////////////////////////////////////////////////// 00004050
C//    THE FIRST PORTION OF THE PROGRAM COMPUTES SECTION LENGTHS      //// 00004060
C//  AND ROTATION ANGLES AND STORES THIS INFORMATION ALONG WITH       //// 00004070
C//  SECTION PROPERTIES IN ARRAY AA WHICH IS PASSED TO FUNCTION       //// 00004080
C//    SUBROUTINE FRDET FOR USE IN THE FORMATION OF THE TRANSFER      //// 00004090
C//                        MATRICES.                                  //// 00004100
C////////////////////////////////////////////////////////////////////////// 00004110
C                                                                           00004120
C    COMPUTE INITIAL UNIT VECTORS                                           00004130
C                                                                           00004140
      B(1)=A(1)                                                             00004150
      B(2)=A(2)                                                             00004160
      B(3)=A(3)+1.D0                                                        00004170
      GO TO 138                                                             00004180
  136 B(1)=PXB-PX(2)                                                        00004190
      B(2)=PYB-PY(2)                                                        00004200
      B(3)=PZB-PZ(2)                                                        00004210
  138 CALL UVEC(A,B,UX,UY,UZ,GAM)                                           00004220
      NIV=1                                                                 00004230
  140 LBR=0                                                                 00004240
      GO TO(999,999,200,300,400,500,500,600,600),NID                        00004250
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004260
C    COMPUTE STRAIGHT SECTION(CHANGE OF PROPERTIES) GEOMETRY                00004270
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004280
  200 DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)                                     00004290
      IF(N.LE.1)GO TO 205                                                   00004300
      IN=N-1                                                                00004310
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL               00004320
```

```
  205 IF(DL.LT.0.D0)GO TO 998                                                    00004330
      IF(DL.EQ.0.D0)GO TO 208                                                    00004340
      AA(N,1)=1.D0                                                               00004350
      AA(N,2)=EY                                                                 00004360
      AA(N,3)=PR                                                                 00004370
      AA(N,4)=RHO                                                                00004380
      AA(N,5)=D1                                                                 00004390
      AA(N,6)=T                                                                  00004400
      AA(N,7)=DL                                                                 00004410
      N=N+1                                                                      00004420
  208 PX(1)=PX(2)                                                                00004430
      PY(1)=PY(2)                                                                00004440
      PZ(1)=PZ(2)                                                                00004450
      IF(LBR-1)100,525,525                                                       00004460
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004470
C    COMPUTE BEND SECTION GEOMETRY                                              00004480
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00004490
  300 IF((N.EQ.1).OR.(NIB.EQ.1))GO TO 302                                        00004500
      B(1)=PXB-PX(2)                                                             00004510
      B(2)=PYB-PY(2)                                                             00004520
      B(3)=PZB-PZ(2)                                                             00004530
      CALL UVEC(A,B,UXP,UYP,UZP,GAM)                                             00004540
      CALL ANGLE(UY,UYP,UX,ALPHA)                                                00004550
      GO TO 303                                                                  00004560
  302 DO 301 I=1,3                                                               00004570
      UXP(I)=UX(I)                                                               00004580
      UYP(I)=UY(I)                                                               00004590
  301 UZP(I)=UZ(I)                                                               00004600
      ALPHA=0.0D0                                                                00004610
  303 ARCL=R*GAM                                                                 00004620
      BL1=BL                                                                     00004630
C                                                                                00004640
C    REORIENT UNIT VECTORS                                                       00004650
C                                                                                00004660
      DO 304 I=1,3                                                               00004670
      UZ(I)=UZP(I)                                                               00004680
      UX(I)=UXP(I)*DCOS(GAM)-UYP(I)*DSIN(GAM)                                    00004690
  304 UY(I)=UXP(I)*DSIN(GAM)+UYP(I)*DCOS(GAM)                                    00004700
      DELTA=PI-GAM                                                               00004710
      BL=R/DTAN(DELTA/2.D0)                                                      00004720
      RA=ARCL/D1                                                                 00004730
C                                                                                00004740
C    DETERMINE THE NUMBER OF SUBSECTIONS OF THE BEND                             00004750
C                                                                                00004760
      IF(MCS.EQ.0)GO TO 305                                                      00004770
      NSS=MCS                                                                    00004780
      GO TO 335                                                                  00004790
  305 IF(RA-1.D0)306,306,310                                                     00004800
```

```
  306 NSS=1                                                              00004810
      GO TO 335                                                          00004820
  310 IF(RA-3.D0)311,311,315                                             00004830
  311 NSS=2                                                              00004840
      GO TO 335                                                          00004850
  315 IF(RA-6.D0)316,316,320                                             00004860
  316 NSS=3                                                              00004870
      GO TO 335                                                          00004880
  320 IF(NMO-6)325,325,326                                               00004890
  325 NM=NMO                                                             00004900
      GO TO 330                                                          00004910
  326 NM=6                                                               00004920
C                                                                        00004930
CC    MAXIMUM NUMBER OF SUBSECTIONS IS 12                                00004940
C                                                                        00004950
  330 NSS=NM*2                                                           00004960
C                                                                        00004970
CC    COMPUTE LENGTH OF STRAIGHT SECTION PRIOR TO BEND                   00004980
C                                                                        00004990
  335 DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)-BL                               00005000
      IF(N.LE.1)GO TO 337                                                00005010
      IN=N-1                                                             00005020
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL1           00005030
  337 IF(DL.LT.0.5D0)GO TO 340                                           00005040
      AA(N,1)=1.D0                                                       00005050
      AA(N,2)=EY                                                         00005060
      AA(N,3)=PR                                                         00005070
      AA(N,4)=RHO                                                        00005080
      AA(N,5)=D1                                                         00005090
      AA(N,6)=T                                                          00005100
      AA(N,7)=DL                                                         00005110
      N=N+1                                                              00005120
      GO TO 345                                                          00005130
C                                                                        00005140
CC    IF LENGTH COMPUTED IS NEGATIVE PRINT AN ERROR MESSAGE              00005150
C                                                                        00005160
  340 IF(DL.LT.-.5D0)GO TO 998                                           00005170
  345 AA(N,1)=2.D0                                                       00005180
      AA(N,2)=EY                                                         00005190
      AA(N,3)=PR                                                         00005200
      AA(N,4)=RHO                                                        00005210
      AA(N,5)=D1                                                         00005220
      AA(N,6)=T                                                          00005230
      AA(N,7)=R                                                          00005240
      AA(N,8)=ALPHA                                                      00005250
      AA(N,9)=GAM                                                        00005260
      AA(N,10)=DFLOAT(NSS)                                               00005270
      N=N+1                                                              00005280
```

```
  355 PX(1)=PX(2)                                                          00005290
      PY(1)=PY(2)                                                          00005300
      PZ(1)=PZ(2)                                                          00005310
      NV=NQ                                                                00005320
      VX=PXB                                                               00005330
      VY=PYB                                                               00005340
      VZ=PZB                                                               00005350
      VR=QR                                                                00005360
      VEY=QEY                                                              00005370
      VPR=QPR                                                              00005380
      VD=QD                                                                00005390
      VT=QT                                                                00005400
      VRHO=QRHO                                                            00005410
      IF(LBR-1)105,530,530                                                 00005420
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00005430
C    COMPUTE CORNER SECTION GEOMETRY                                       00005440
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00005450
  400 IF((N.EQ.1).OR.(NIB.EQ.1))GO TO 402                                  00005460
      B(1)=PXB-PX(2)                                                       00005470
      B(2)=PYB-PY(2)                                                       00005480
      B(3)=PZB-PZ(2)                                                       00005490
      CALL UVEC(A,B,UXP,UYP,UZP,GAM)                                       00005500
      CALL ANGLE(UY,UYP,UX,ALPHA)                                          00005510
      GO TO 403                                                            00005520
  402 DO 401 I=1,3                                                         00005530
      UXP(I)=UX(I)                                                         00005540
      UYP(I)=UY(I)                                                         00005550
  401 UZP(I)=UZ(I)                                                         00005560
      ALPHA=0.0D0                                                          00005570
C                                                                          00005580
C    REORIENT UNIT VECTORS                                                 00005590
C                                                                          00005600
  403 DO 410 I=1,3                                                         00005610
      UZ(I)=UZP(I)                                                         00005620
      UX(I)=UXP(I)*DCOS(GAM)-UYP(I)*DSIN(GAM)                              00005630
  410 UY(I)=UXP(I)*DSIN(GAM)+UYP(I)*DCOS(GAM)                              00005640
C                                                                          00005650
C    COMPUTE PRE-CORNER STRAIGHT SECTION LENGTH                            00005660
C                                                                          00005670
  411 DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)                                    00005680
      IF(N.LE.1)GO TO 415                                                  00005690
      IN=N-1                                                               00005700
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL              00005710
C                                                                          00005720
C    IF LENGTH COMPUTED IS NEGATIVE PRINT AN ERROR MESSAGE                 00005730
C                                                                          00005740
  415 IF(DL.LE.0.D0)GO TO 998                                              00005750
      AA(N,1)=3.D0                                                         00005760
```

```
      AA(N,2)=EY                                                        00005770
      AA(N,3)=PR                                                        00005780
      AA(N,4)=RHO                                                       00005790
      AA(N,5)=D1                                                        00005800
      AA(N,6)=T                                                         00005810
      AA(N,7)=DL                                                        00005820
      AA(N,8)=ALPHA                                                     00005830
      AA(N,9)=GAM                                                       00005840
      N=N+1                                                             00005850
      PX(1)=PX(2)                                                       00005860
      PY(1)=PY(2)                                                       00005870
      PZ(1)=PZ(2)                                                       00005880
      VX=PXB                                                            00005890
      VY=PYB                                                            00005900
      VZ=PZB                                                            00005910
      VR=QR                                                             00005920
      VEY=QEY                                                           00005930
      VPR=QPR                                                           00005940
      VD=QD                                                             00005950
      VT=QT                                                             00005960
      VRHO=QRHO                                                         00005970
      NV=NQ                                                             00005980
      IF(LBR-1)105,530,530                                              00005990
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00006000
C     COMPUTE BRANCH OR DUAL-BRANCH GEOMETRY                            00006010
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00006020
  500 LBR=1                                                             00006030
      DO 502 I=1,3                                                      00006040
      SX(I)=UX(I)                                                       00006050
      SY(I)=UY(I)                                                       00006060
  502 SZ(I)=UZ(I)                                                       00006070
C                                                                       00006080
CC    COMPUTE LENGTH OF STRAIGHT SECTION PRECEEDING JUNCTION            00006090
C                                                                       00006100
      DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)                                 00006110
      IF(N.LE.1)GO TO 503                                               00006120
      IN=N-1                                                            00006130
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL           00006140
C                                                                       00006150
CC    IF LENGTH COMPUTED IS NEGATIVE PRINT AN ERROR MESSAGE             00006160
C                                                                       00006170
  503 IF(DL.LE.0.D0)GO TO 998                                           00006180
      AA(N,2)=EY                                                        00006190
      AA(N,3)=PR                                                        00006200
      AA(N,4)=RHO                                                       00006210
      AA(N,5)=D1                                                        00006220
      AA(N,6)=T                                                         00006230
      AA(N,7)=DL                                                        00006240
```

```
      IF(NID.EQ.7)GO TO 505                                                    00006250
      AA(N,1)=5.DO                                                             00006260
      KB=1                                                                     00006270
      GO TO 510                                                                00006280
  505 AA(N,1)=6.DO                                                             00006290
      KB=0                                                                     00006300
  510 N=N+1                                                                    00006310
  520 K=K+1                                                                    00006320
C                                                                              00006330
CC    READ BRANCH START POINT                                                  00006340
C                                                                              00006350
      READ(5,61)NID,PX(1),PY(1),PZ(1),JBC                                      00006360
      WRITE(6,82)NID,PX(1),PY(1),PZ(1),JBC                                     00006370
      NB(K)=JBC                                                                00006380
C                                                                              00006390
CC    READ NEXT POINT AND SECTION PROPERTIES                                   00006400
C                                                                              00006410
  525 READ(5,62)NV,VX,VY,VZ,VR,VEY,VPR,VD,VT,VRHO                              00006420
      WRITE(6,83)NV,VX,VY,VZ,VR,VEY,VPR,VD,VT,VRHO                             00006430
  530 IF((NV.NE.4).AND.(NV.NE.5).AND.(NV.NE.8))GO TO 535                       00006440
C                                                                              00006450
CC    IF END,CORNER,OR BEND POINT READ FOLLOWING POINT                         00006460
C                                                                              00006470
      READ(5,62)NQ,PXB,PYB,PZB,QR,QEY,QPR,QD,QT,QRHO                           00006480
      WRITE(6,83)NQ,PXB,PYB,PZB,QR,QEY,QPR,QD,QT,QRHO                          00006490
  535 NID=NV                                                                   00006500
      PX(2)=VX                                                                 00006510
      PY(2)=VY                                                                 00006520
      PZ(2)=VZ                                                                 00006530
      R=VR                                                                     00006540
      IF(VEY.GT.1.D-9)EY=VEY                                                   00006550
      IF(VPR.GT.1.D-9)PR=VPR                                                   00006560
      IF(VD.GT.1.D-9)D1=VD                                                     00006570
      IF(VT.GT.1.D-9)T=VT                                                      00006580
      IF(VRHO.GT.1.D-9)RHO=VRHO                                                00006590
      A(1)=PX(2)-PX(1)                                                         00006600
      A(2)=PY(2)-PY(1)                                                         00006610
      A(3)=PZ(2)-PZ(1)                                                         00006620
  560 IF(NIB.GT.0)GO TO 565                                                    00006630
      IF((NID.EQ.4).OR.(NID.EQ.5).OR.(NID.EQ.8))GO TO 561                      00006640
C                                                                              00006650
CC    COMPUTE INITIAL UNIT VECTORS                                             00006660
C                                                                              00006670
      B(1)=A(1)                                                                00006680
      B(2)=A(2)                                                                00006690
      B(3)=A(3)+1.DO                                                           00006700
      GO TO 563                                                                00006710
  561 B(1)=PXB-PX(2)                                                           00006720
```

```
      B(2)=PYB-PY(2)                                                    00006730
      B(3)=PZB-PZ(2)                                                    00006740
  563 CALL UVEC(A,B,UX,UY,UZ,GAM)                                       00006750
  565 NIB=NIB+1                                                         00006760
C                                                                       00006770
CC    COMPUTE APPROPRIATE SECTION GEOMETRY USING MAIN MEMBER ROUTINES   00006780
C                                                                       00006790
  566 GO TO(999,999,200,300,400),NID                                    00006800
C                                                                       00006810
CC    COMPUTE BRANCH JUNCTION SECTION GEOMETRY                          00006820
C                                                                       00006830
  570 DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)                                 00006840
      IN=N-1                                                            00006850
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL           00006860
  571 IF(DL.LE.0.D0)GO TO 998                                           00006870
      AA(N,1)=7.D0                                                      00006880
      AA(N,2)=EY                                                        00006890
      AA(N,3)=PR                                                        00006900
      AA(N,4)=RHO                                                       00006910
      AA(N,5)=D1                                                        00006920
      AA(N,6)=T                                                         00006930
      AA(N,7)=DL                                                        00006940
      B(1)=PXB-PX(2)                                                    00006950
      B(2)=PYB-PY(2)                                                    00006960
      B(3)=PZB-PZ(2)                                                    00006970
      CALL UVEC(A,B,UXP,UYP,UZP,GAM)                                    00006980
C                                                                       00006990
CC    COMPUTE INITIAL X-AXIS ROTATION ANGLE OF BRANCH UNIT VECTORS      00007000
C                                                                       00007010
      CALL ANGLE(UY,UYP,UX,AL1)                                         00007020
      AA(N,8)=AL1                                                       00007030
C                                                                       00007040
CC    COMPUTE Z-AXIS ROTATION TO ALIGN X COMPONENTS OF MAIN AND BRANCH  00007050
CC      UNIT VECTORS                                                    00007060
C                                                                       00007070
      CALL ANGLE(UXP,SX,UZP,GAM)                                        00007080
      AA(N,9)=GAM                                                       00007090
      GAM1=DABS(GAM)                                                    00007100
      DO 575 I=1,3                                                      00007110
      UZDP(I)=UZP(I)                                                    00007120
      UXDP(I)=UXP(I)*DCOS(GAM1)-UYP(I)*DSIN(GAM1)                       00007130
  575 UYDP(I)=UXP(I)*DSIN(GAM1)+UYP(I)*DCOS(GAM1)                       00007140
C                                                                       00007150
CC    COMPUTE FINAL X-AXIS ROTATION TO ALIGN Y AND Z COMPONENTS OF MAIN 00007160
CC      AND BRANCH UNIT VECTORS                                         00007170
C                                                                       00007180
      CALL ANGLE(UYDP,SY,SX,AL)                                         00007190
      AA(N,10)=AL                                                       00007200
```

```
      N=N+1                                                                  00007210
      KB=KB+1                                                                00007220
      NIB=0                                                                  00007230
C                                                                            00007240
C    IF DUAL-BRANCH REPEAT BRANCH GEOMETRY ROUTINES                          00007250
C                                                                            00007260
      IF(KB.EQ.1)GO TO 520                                                   00007270
      DO 580 I=1,3                                                           00007280
      UX(I)=SX(I)                                                            00007290
      UY(I)=SY(I)                                                            00007300
  580 UZ(I)=SZ(I)                                                            00007310
      PX(1)=PX(2)                                                            00007320
      PY(1)=PY(2)                                                            00007330
      PZ(1)=PZ(2)                                                            00007340
      VX=PXB                                                                 00007350
      VY=PYB                                                                 00007360
      VZ=PZB                                                                 00007370
      VR=QR                                                                  00007380
      VEY=QEY                                                                00007390
      VPR=QPR                                                                00007400
      VD=QD                                                                  00007410
      VT=QT                                                                  00007420
      VRHO=QRHO                                                              00007430
      NV=NQ                                                                  00007440
      GO TO 105                                                              00007450
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00007460
C    COMPUTE END SECTION GEOMETRY                                            00007470
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00007480
  600 DL=DSQRT(A(1)**2+A(2)**2+A(3)**2)                                      00007490
      IF(N.LE.1)GO TO 604                                                    00007500
      IN=N-1                                                                 00007510
      IF((AA(IN,1).GT.1.9D0).AND.(AA(IN,1).LT.2.1D0))DL=DL-BL                00007520
  604 IF(DL.LT.-.5D0)GO TO 998                                               00007530
      IF((DL.LT.0.1D0).AND.(DL.GE.-.5D0))DL=0.D0                             00007540
      AA(N,1)=4.D0                                                           00007550
      AA(N,2)=EY                                                             00007560
      AA(N,3)=PR                                                             00007570
      AA(N,4)=RHO                                                            00007580
      AA(N,5)=D1                                                             00007590
      AA(N,6)=T                                                              00007600
      AA(N,7)=DL                                                             00007610
  605 FORMAT(6X,'PROPERTIES AND GEOMETRY:',//)                               00007620
C                                                                            00007630
C    PRINT ARRAY OF PROPERTIES AND GEOMETRY                                  00007640
C                                                                            00007650
  610 WRITE(6,85)                                                            00007660
      WRITE(6,605)                                                           00007670
      WRITE(6,620)                                                           00007680
```

```
  620 FORMAT(6X,'SECT',8X,'ELASTIC',4X,'POISSONS',19X,'WALL',5X,'LENGTH 00007690
     1OR',24X,'NR OF SUBSEC',/,7X,'NR',3X,'ID',4X,'MODULUS',6X,'RATIO',300007700
     2X,' WT/VOL ',2X,'O.D.',3X,'THICK',4X,'RAD OF CURV',4X,'ALPHA1 ',3X00007710
     3,'GAMMA',4X,'OR ALPHA',/)                                        00007720
      DO 630 I=1,N                                                     00007730
  630 WRITE(6,635)I,(AA(I,J),J=1,10)                                   00007740
  635 FORMAT(7X,I2,3X,F3.1,2X,E9.3,5X,F3.2,6X,F6.1,2X,F5.2,3X,F6.4,4X,  00007750
     1F8.3,5X,F6.3,4X,F6.3,5X,F6.3)                                    00007760
      WRITE(6,85)                                                      00007770
  636 FORMAT(6X,'BOUNDARY CONDITION CODES:',//)                        00007780
  637 FORMAT(6X,'LEFT B.C. CODE=',1X,I1)                               00007790
  638 FORMAT(6X,'RIGHT B.C. CODE=',I1)                                 00007800
      WRITE(6,636)                                                     00007810
      WRITE(6,637)NB(1)                                                00007820
      WRITE(6,638)NB(2)                                                00007830
      IF(K.LT.3)GO TO 1000                                             00007840
      DO 640 I=3,K                                                     00007850
      JI=I-2                                                           00007860
  640 WRITE(6,642)JI,NB(I)                                             00007870
  642 FORMAT(6X,'BRANCH #',I2,1X,'B.C. CODE=',1X,I1)                   00007880
      GO TO 1000                                                       00007890
C                                                                      00007900
C   THE FOLLOWING ARE THE ERROR MESSAGES                               00007910
C                                                                      00007920
  998 WRITE(6,995)N                                                    00007930
      GO TO 1001                                                       00007940
  999 WRITE(6,996)N                                                    00007950
      GO TO 1001                                                       00007960
  995 FORMAT(//,6X,'GEOMETRY ERROR-RECOMPUTE WORKING POINTS IN SECTION',00007970
     1I4)                                                              00007980
  996 FORMAT(//,6X,'POINT IDENTIFICATION NUMBER HAD A STARTING POINT COD00007990
     1E AT SECTION',I4)                                                00008000
 1000 CONTINUE                                                         00008010
C                                                                      00008020
C////////////////////////////////////////////////////////////////////// 00008030
C//     THE SECOND PORTION OF THE MAIN PROGRAM COMPUTES SUCCESSIVE  //// 00008040
C//  VALUES OF THE FREQUENCY DETERMINANT AND ITERATES UNTIL A       //// 00008050
C//  NATURAL FREQUENCY OF THE PIPING SYSTEM IS LOCATED .            //// 00008060
C////////////////////////////////////////////////////////////////////// 00008070
C                                                                      00008080
      EYT=0.D0                                                         00008090
      PRT=0.D0                                                         00008100
      RHOT=0.D0                                                        00008110
      D1T=0.D0                                                         00008120
      TT=0.D0                                                          00008130
      DLT=0.0D0                                                        00008140
      NN=0                                                             00008150
C                                                                      00008160
```

```
C   CONSTRUCT SYNTHETIC STRAIGHT PIPE EQUAL IN LENGTH TO THE MAIN       00008170
C      MEMBER AND HAVING THE AVERAGE VALUES OF THE PROPERTIES OF THE    00008180
C      MAIN MEMBER SECTIONS                                             00008190
C                                                                       00008200
  700 NN=NN+1                                                           00008210
      NSEC=AA(NN,1)                                                     00008220
      IF(NSEC.EQ.2)GO TO 705                                            00008230
      DL=AA(NN,7)                                                       00008240
      GO TO 706                                                         00008250
  705 DL=AA(NN,7)*AA(NN,9)                                              00008260
  706 DLT=DLT+DL                                                        00008270
      EYT=EYT+AA(NN,2)*DL                                               00008280
      PRT=PRT+AA(NN,3)*DL                                               00008290
      RHOT=RHOT+AA(NN,4)*DL                                             00008300
      D1T=D1T+AA(NN,5)*DL                                               00008310
      TT=TT+AA(NN,6)*DL                                                 00008320
  707 GO TO(700,700,700,714,711,713),NSEC                               00008330
  711 NN=NN+1                                                           00008340
      NSEC=AA(NN,1)                                                     00008350
      IF(NSEC.EQ.7)GO TO 700                                            00008360
      GO TO 711                                                         00008370
  713 NN=NN+1                                                           00008380
      NSEC=AA(NN,1)                                                     00008390
      IF(NSEC.EQ.7)GO TO 711                                            00008400
      GO TO 713                                                         00008410
  714 DEN=DLT                                                           00008420
      EY=EYT/DEN                                                        00008430
      PR=PRT/DEN                                                        00008440
      RHO=RHOT/DEN                                                      00008450
      D1=D1T/DEN                                                        00008460
      T=TT/DEN                                                          00008470
      DL=DLT                                                            00008480
C                                                                       00008490
C   FIND THE FUNDAMENTAL FREQUENCY OF THE SYNTHETIC STRAIGHT PIPE       00008500
C                                                                       00008510
 7140 W1=.1D0                                                           00008520
      CALL WSTART(W1,DL,RHO,EY,PR,D1,T,MSR,1,DW1)                       00008530
  715 W2=W1+5.D0                                                        00008540
      CALL WSTART(W2,DL,RHO,EY,PR,D1,T,MSR,1,DW2)                       00008550
      NDISCR=DW1/DABS(DW1)+DW2/DABS(DW2)                                00008560
  716 IF(NDISCR.EQ.0)GO TO 717                                          00008570
      W1=W2                                                             00008580
      DW1=DW2                                                           00008590
      GO TO 715                                                         00008600
  717 X1=(DABS(DW1)*(W2-W1))/(DABS(DW1)+DABS(DW2))                      00008610
C                                                                       00008620
C   THE STARTING FREQUENCY INCREMENT IS 1/6 OF THE 1ST SYNTHETIC        00008630
C      FREQUENCY                                                        00008640
```

```
C                                                                       00008650
      WINCR=(W1+X1)/6.D0                                                00008660
      FINCR=WINCR                                                       00008670
      NM=1                                                              00008680
      WRITE(6,85)                                                       00008690
      WRITE(6,712)                                                      00008700
  712 FORMAT(6X,'PIPING SYSTEM NATURAL FREQUENCIES:',//)                00008710
  719 FORMAT(6X,'MODE',3X,'FREQUENCY(RAD/SEC)',3X,'FREQUENCY DETERMINANT00008720
     1',/)                                                              00008730
  720 FORMAT(7X,I2,6X,F13.7,8X,E11.5)                                   00008740
      WRITE(6,719)                                                      00008750
      W1=.10D0                                                          00008760
      DW1=FRDET(W1)                                                     00008770
  718 X(1)=W1                                                           00008780
      Y(1)=DW1                                                          00008790
      YMAX=DW1                                                          00008800
      YMIN=DW1                                                          00008810
      XMIN=0.D0                                                         00008820
      M=2                                                               00008830
C                                                                       00008840
C   INCREMENT FREQUENCY UNTIL THE SIGN OF THE FREQUENCY DETERMINANT     00008850
C      CHANGES                                                          00008860
C                                                                       00008870
  721 W2=W1+WINCR                                                       00008880
      XMAX=W2                                                           00008890
      DW2=FRDET(W2)                                                     00008900
      DW2CHK=DABS(DW2)                                                  00008910
      IF(DW2CHK.LT.1.D-60)GO TO 756                                     00008920
      IF(DW2CHK.GT.1.D+60)GO TO 756                                     00008930
      IF(DW2.LT.YMIN)YMIN=DW2                                           00008940
      IF(DW2.GT.YMAX)YMAX=DW2                                           00008950
  722 X(M)=W2                                                           00008960
      Y(M)=DW2                                                          00008970
C                                                                       00008980
C   CHECK TO SEE IF THE SIGN OF THE FREQUENCY DETERMINANT HAS CHANGED   00008990
C                                                                       00009000
      NDISCR=DW1/DABS(DW1)+DW2/DABS(DW2)                                00009010
      M=M+1                                                             00009020
      IF(NDISCR.EQ.0)GO TO 724                                          00009030
      IF(M.EQ.1000)GO TO 751                                            00009040
      W1=W2                                                             00009050
      DW1=DW2                                                           00009060
      GO TO 721                                                         00009070
C                                                                       00009080
C   THE SOLUTION ACCEPTABILITY CRITERION IS EQUAL TO THE INITIAL        00009090
C     FREQUENCY INCREMENT DIVIDED BY TEN THOUSAND                       00009100
C                                                                       00009110
  724 EPS=FINCR/1.D4                                                    00009120
```

```
C                                                                       00009130
C   ONCE A ROOT HAS BEEN BRACKETTED BY INCREMENTING FREQUENCY UNTIL     00009140
C     THE SIGN OF THE FREQUENCY DETERMINANT CHANGES, ITERATION IS       00009150
C     ACCOMPLISHED BY MUELLER'S METHOD UNTIL THE ROOT IS LOCATED        00009160
C                                                                       00009170
      CALL DROOT(WN,DWN,FRDET,W1,W2,EPS,NIT,IER)                        00009180
      IF(IER-1)726,725,750                                              00009190
  725 WRITE(6,727)                                                      00009200
  727 FORMAT('+',50X,'ITERATION OF THE NEXT FREQUENCY VALUE DID NOT CONV00009210
     1ERGE IN SUBROUTINE DROOT')                                        00009220
  726 FN(NM)=WN                                                         00009230
      WRITE(6,720)NM,FN(NM),DWN                                         00009240
      NM=NM+1                                                           00009250
      IF(NM-NMO)730,730,735                                             00009260
  730 IF(NM.LE.2)GO TO 733                                              00009270
C                                                                       00009280
C   AFTER THE SECOND MODE FREQUENCY IS FOUND CHANGE THE FREQUENCY       00009290
C     INCREMENT TO 1/10 THE DIFFERENCE BETWEEN THE LAST TWO MODES       00009300
C                                                                       00009310
      WINCR=(FN(NM-1)-FN(NM-2))/10.D0                                   00009320
  733 W1=WN+1.D0                                                        00009330
      DW1=FRDET(W1)                                                     00009340
      X(M)=WN                                                           00009350
      Y(M)=DWN                                                          00009360
      M=M+1                                                             00009370
      GO TO 721                                                         00009380
  735 X(M)=WN                                                           00009390
      Y(M)=DWN                                                          00009400
C                                                                       00009410
C   SORT THE ARRAY OF VALUES OF FREQUENCIES AND FREQUENCY DETERMINANT   00009420
C     FOR PLOTTING ON THE PRINTER                                       00009430
C                                                                       00009440
  740 CALL SORT(X,Y,M)                                                  00009450
  742 FORMAT('1',6X,'GRAPH OF THE VALUE OF FREQUENCY DETERMINANT VS. FRE00009460
     1QUENCY',///)                                                      00009470
      WRITE(6,742)                                                      00009480
C                                                                       00009490
C   PLOT THE ARRAY OF VALUES OF FREQUENCY DETERMINANT  VS. FREQUENCY    00009500
C                                                                       00009510
      CALL PRPLOT(X,Y,M,XMIN,XMAX,YMIN,YMAX,0)                          00009520
      GO TO 751                                                         00009530
  750 WRITE(6,755)                                                      00009540
  755 FORMAT(///,6X,'PROBLEM TERMINATED,DW1/DW2 HAD SAME SIGN GOING INTO00009550
     1 DROOT',//)                                                       00009560
      GO TO 740                                                         00009570
  756 WRITE(6,757)                                                      00009580
  757 FORMAT(///,6X,'PROBLEM TERMINATED, SIGNIFICANT FIGURE LIMITATION O00009590
     1F THE COMPUTER WAS EXCEEDED',///)                                 00009600
```

```
      GO TO 740                                                          00009610
  751 NPROB1=NPROB1+1                                                    00009620
      IF(NPROB-NPROB1)1001,23,23                                         00009630
 1001 STOP                                                               00009640
      END                                                                00009650

      SUBROUTINE ANGLE(UY,UYP,UX,ALPHA)                                  00009660
C*************************************************************          00009670
C********          SUBROUTINE ANGLE                           **********  00009680
C********     COMPUTES THE ANGLE OF ROTATION                  **********  00009690
C********   ABOUT A COMMON AXIS OF TWO SETS OF                **********  00009700
C********            UNIT VECTORS                             **********  00009710
C*************************************************************          00009720
C     UY IS INITIAL UNIT VECTOR                                          00009730
C     UYP IS FINAL UNIT VECTOR AFTER ROTATION THRU ALPHA                 00009740
C     UX IS UNIT VECTOR USED TO CHECK THE SIGN OF ALPHA                  00009750
      IMPLICIT REAL*8(A-H,O-Z)                                           00009760
      DIMENSION UY(3),UYP(3),UX(3),C(3)                                  00009770
      DOTPR=(UY(1)*UYP(1)+UY(2)*UYP(2)+UY(3)*UYP(3))                     00009780
      ALPHA=DARCOS(DOTPR)                                                00009790
      C(1)=(UY(2)*UYP(3)-UYP(2)*UY(3))                                   00009800
      C(2)=(UY(3)*UYP(1)-UYP(3)*UY(1))                                   00009810
      C(3)=(UY(1)*UYP(2)-UYP(1)*UY(2))                                   00009820
      TRIPR=(C(1)*UX(1)+C(2)*UX(2)+C(3)*UX(3))                           00009830
      IF(TRIPR.LT.0.0D0)ALPHA=-ALPHA                                     00009840
      RETURN                                                             00009850
      END                                                                00009860

      SUBROUTINE COORDX(AL,N,CTX)                                        00009870
C*************************************************************          00009880
C********          SUBROUTINE COORDX                          **********  00009890
C********    COMPUTES COORDINATE TRANSFORMATION               **********  00009900
C********      MATRIX FOR ROTATION ABOUT THE                  **********  00009910
C********              X-AXIS                                 **********  00009920
C*************************************************************          00009930
      IMPLICIT REAL*8(A-H,O-Z)                                           00009940
      DIMENSION CX(12,12),CTX(N,N)                                       00009950
      DO 10 I=1,12                                                       00009960
      DO 10 J=1,12                                                       00009970
   10 CX(I,J)=0.0D0                                                      00009980
      A=DCOS(AL)                                                         00009990
      B=DSIN(AL)                                                         00010000
      CX(1,1)=1.D0                                                       00010010
      CX(2,2)=A                                                          00010020
      CX(2,3)=B                                                          00010030
      CX(3,2)=-B                                                         00010040
```

```
      CX(3,3)=A                                                         00010050
      CX(4,4)=1.D0                                                      00010060
      CX(5,5)=A                                                         00010070
      CX(5,6)=B                                                         00010080
      CX(6,5)=-B                                                        00010090
      CX(6,6)=A                                                         00010100
      CX(7,7)=1.D0                                                      00010110
      CX(8,8)=A                                                         00010120
      CX(8,9)=B                                                         00010130
      CX(9,8)=-B                                                        00010140
      CX(9,9)=A                                                         00010150
      CX(10,10)=1.D0                                                    00010160
      CX(11,11)=A                                                       00010170
      CX(11,12)=B                                                       00010180
      CX(12,11)=-B                                                      00010190
      CX(12,12)=A                                                       00010200
      DO 20 I=1,N                                                       00010210
      DO 20 J=1,N                                                       00010220
   20 CTX(I,J)=CX(I,J)                                                  00010230
      RETURN                                                            00010240
      END                                                               00010250

      SUBROUTINE COORDZ(GAM,N,CTZ)                                      00010260
C*************************************************************        00010270
C********          SUBROUTINE COORDZ                          **********00010280
C********   COMPUTES COORDINATE TRANSFORMATION                **********00010290
C********     MATRIX FOR ROTATION ABOUT THE                   **********00010300
C********           Z-AXIS                                    **********00010310
C*************************************************************        00010320
      IMPLICIT REAL*8(A-H,O-Z)                                          00010330
      DIMENSION CZ(12,12),CTZ(N,N)                                      00010340
      DO 10 I=1,12                                                      00010350
      DO 10 J=1,12                                                      00010360
   10 CZ(I,J)=0.0D0                                                     00010370
      A=DCOS(GAM)                                                       00010380
      B=DSIN(GAM)                                                       00010390
      CZ(1,1)=A                                                         00010400
      CZ(1,2)=B                                                         00010410
      CZ(2,1)=-B                                                        00010420
      CZ(2,2)=A                                                         00010430
      CZ(3,3)=1.D0                                                      00010440
      CZ(4,4)=A                                                         00010450
      CZ(4,5)=B                                                         00010460
      CZ(5,4)=-B                                                        00010470
      CZ(5,5)=A                                                         00010480
      CZ(6,6)=1.D0                                                      00010490
      CZ(7,7)=A                                                         00010500
```

```
      CZ(7,8)=B                                                          00010510
      CZ(8,7)=-B                                                         00010520
      CZ(8,8)=A                                                          00010530
      CZ(9,9)=1.D0                                                       00010540
      CZ(10,10)=A                                                        00010550
      CZ(10,11)=B                                                        00010560
      CZ(11,10)=-B                                                       00010570
      CZ(11,11)=A                                                        00010580
      CZ(12,12)=1.D0                                                     00010590
      DO 20 I=1,N                                                        00010600
      DO 20 J=1,N                                                        00010610
   20 CTZ(I,J)=CZ(I,J)                                                   00010620
      RETURN                                                             00010630
      END                                                                00010640

      SUBROUTINE CURMAT(EY,PR,DO,R,T,GAM,MSR,UCV)                        00010650
C*************************************************************         00010660
C********          SUBROUTINE CURMAT                  **********         00010670
C********    CONSTRUCTS A FIELD MATRIX FOR A          **********         00010680
C********          CURVED SECTION OF PIPE             **********         00010690
C*************************************************************         00010700
      IMPLICIT REAL*8(A-H,O-Z)                                           00010710
      DIMENSION UCV(12,12)                                               00010720
      PI=3.1415926535897 93                                              00010730
      DI=DO-2.D0*T                                                       00010740
      RM=(DO-T)/2.D0                                                     00010750
      RGYRS=(DO**2+DI**2)/16.D0                                          00010760
      DJ=RGYRS*PI*(DO**2-DI**2)/4.D0                                     00010770
      V=1.D0-2.D0*T/DO                                                   00010780
      IF(MSR.EQ.0)GO TO 10                                               00010790
      SDF=((7.D0+14.D0*PR+8.D0*PR**2)+((2.D0*V**2)/(1.D0+V**2)**2)       00010800
     1*(10.D0+20.D0*PR+8.D0*PR**2))/(6.D0*(1.D0+PR)**2)                  00010810
      GO TO 20                                                           00010820
   10 SDF=0.0D0                                                          00010830
   20 XL=2.D0*(1.D0+PR)*SDF                                              00010840
      BFF=(1.65D0*RM**2)/(R*T)                                           00010850
      IF(BFF.LT.1.D0)BFF=1.D0                                            00010860
      C=DCOS(GAM/2.D0)                                                   00010870
      S=DSIN(GAM/2.D0)                                                   00010880
      CO=DCOS(GAM)                                                       00010890
      SO=DSIN(GAM)                                                       00010900
      CC=C**2                                                            00010910
      SS=S**2                                                            00010920
      CO2=CO**2                                                          00010930
      SO2=SO**2                                                          00010940
      P=(R**2*BFF+RGYRS*(1.D0-XL))/2.D0                                  00010950
      Q=(1.D0+PR-BFF)/2.D0                                               00010960
```

```
      PS1=GAM+S0                                                        00010970
      PS2=GAM-S0                                                        00010980
      ZZ=R/(EY*DJ)                                                      00010990
      A=(RGYRS*XL*GAM+P*PS1)*ZZ                                         00011000
      B=(RGYRS*XL*GAM+P*PS2)*ZZ                                         00011010
      XM=(RGYRS*XL+R**2*(1.D0+PR))*GAM*ZZ                               00011020
      D=(2.D0*R*(1.D0+PR)*S)*ZZ                                         00011030
      E=(BFF*GAM+Q*PS1)*ZZ                                              00011040
      F=(BFF*GAM+Q*PS2)*ZZ                                              00011050
      G=BFF*GAM*ZZ                                                      00011060
      H=-2.D0*R*BFF*S*ZZ                                                00011070
      A1=A*CC+2.D0*R*C*H+SS*B+R**2*G                                    00011080
      A2=-A*C*S+B*C*S-R*S*H                                             00011090
      A3=-C*H-R*G                                                       00011100
      B2=A*SS+B*CC                                                      00011110
      C1=XM-2.D0*R*C*D+R**2*CC*E+R**2*SS*F                              00011120
      C2=-C*D+R*CC*E+R*SS*F                                             00011130
      C3=S*D+R*S0*(F-E)/2.D0                                            00011140
      D1=-C*D+R*CC*E+R*SS*F                                             00011150
      D2=CC*E+SS*F                                                      00011160
      D3=S0*(F-E)/2.D0                                                  00011170
      E3=SS*E+CC*F                                                      00011180
      COM1=C0-1.D0                                                      00011190
      DO 30 I=1,12                                                      00011200
      DO 30 J=1,12                                                      00011210
   30 UCV(I,J)=0.0D0                                                    00011220
      UCV(1,1)=C0                                                       00011230
      UCV(1,2)=-S0                                                      00011240
      UCV(1,6)=R*COM1                                                   00011250
      UCV(1,7)=A1*C0+A3*R*(COM1)-A2*S0                                  00011260
      UCV(1,8)=A2*C0+S*H*R*(COM1)-A2*S0                                 00011270
      UCV(1,12)=A3*C0+G*R*(COM1)-S0*S*H                                 00011280
      UCV(2,1)=S0                                                       00011290
      UCV(2,2)=C0                                                       00011300
      UCV(2,6)=R*S0                                                     00011310
      UCV(2,7)=A1*S0+A3*R*S0+A2*C0                                      00011320
      UCV(2,8)=A2*S0+S*H*R*S0+A2*C0                                     00011330
      UCV(2,12)=A3*S0+G*R*S0+C0*S*H                                     00011340
      UCV(3,3)=1.D0                                                     00011350
      UCV(3,4)=UCV(1,6)                                                 00011360
      UCV(3,5)=-R*S0                                                    00011370
      UCV(3,9)=C1+D1*R*(COM1)-C3*R*S0                                   00011380
      UCV(3,10)=C2+D2*R*(COM1)-D3*R*S0                                  00011390
      UCV(3,11)=C3+D3*R*(COM1)-E3*R*S0                                  00011400
      UCV(4,4)=C0                                                       00011410
      UCV(4,5)=-S0                                                      00011420
      UCV(4,9)=C0*D1-S0*C3                                              00011430
      UCV(4,10)=C0*D2-S0*D3                                             00011440
```

```
      UCV(4,11)=C0*D3-S0*E3                                              00011450
      UCV(5,4)=S0                                                        00011460
      UCV(5,5)=C0                                                        00011470
      UCV(5,9)=S0*D1+C0*C3                                               00011480
      UCV(5,10)=S0*D2+C0*D3                                              00011490
      UCV(5,11)=S0*D3+C0*E3                                              00011500
      UCV(6,6)=1.D0                                                      00011510
      UCV(6,7)=A3                                                        00011520
      UCV(6,8)=S*H                                                       00011530
      UCV(6,12)=G                                                        00011540
      UCV(7,7)=C0                                                        00011550
      UCV(7,8)=-S0                                                       00011560
      UCV(8,7)=S0                                                        00011570
      UCV(8,8)=C0                                                        00011580
      UCV(9,9)=1.D0                                                      00011590
      UCV(10,9)=UCV(1,6)                                                 00011600
      UCV(10,10)=C0                                                      00011610
      UCV(10,11)=-S0                                                     00011620
      UCV(11,9)=R*S0                                                     00011630
      UCV(11,10)=S0                                                      00011640
      UCV(11,11)=C0                                                      00011650
      UCV(12,7)=UCV(1,6)                                                 00011660
      UCV(12,8)=-R*S0                                                    00011670
      UCV(12,12)=1.D0                                                    00011680
   50 RETURN                                                             00011690
      END                                                                00011700


      SUBROUTINE DETER(N,A,D)                                            00011710
C*************************************************************          00011720
C********          SUBROUTINE DETER                           **********  00011730
C********    COMPUTES THE VALUE OF THE                        **********  00011740
C********     FREQUENCY DETERMINANT                           **********  00011750
C*************************************************************          00011760
      IMPLICIT REAL*8 (A-H), REAL*8 (O-Z)                                00011770
      DIMENSION A(N,N)                                                   00011780
      DD=1.D0                                                            00011790
   10  DO 34 L=1,N                                                       00011800
       KP=0                                                              00011810
       Z=0.0                                                             00011820
       DO 12 K=L,N                                                       00011830
      IF(Z-DABS(A(K,L)))11,12,12                                         00011840
   11 Z=DABS(A(K,L))                                                     00011850
       KP=K                                                              00011860
   12  CONTINUE                                                          00011870
       IF(L-KP)13,20,20                                                  00011880
   13  DO 14 J=L,N                                                       00011890
       Z=A(L,J)                                                          00011900
```

```
      A(L,J)=A(KP,J)                                                    00011910
   14 A(KP,J)=Z                                                         00011920
      DD=-DD                                                            00011930
   20 IF(L-N)31,40,40                                                   00011940
   31 LP1=L+1                                                           00011950
      DO 34 K=LP1,N                                                     00011960
      IF(A(K,L))32,34,32                                                00011970
   32 RATIO=A(K,L)/A(L,L)                                               00011980
      DO 33 J=LP1,N                                                     00011990
   33 A(K,J)=A(K,J)-RATIO*A(L,J)                                        00012000
   34 CONTINUE                                                          00012010
   40 DO 41 K = 1,N                                                     00012020
   41 DD=DD*A(K,K)                                                      00012030
      D=DD                                                              00012040
      RETURN                                                            00012050
      END                                                               00012060


      SUBROUTINE DROOT(X,F,FCT,XLI,XRI,EPS,IEND,IER)                    00012070
C*************************************************************        00012080
C********          SUBROUTINE DROOT                         **********  00012090
C********     ITERATES USING MUELLER'S METHOD TO            **********  00012100
C********     FIND THE VALUE OF THE NATURAL                 **********  00012110
C********          FREQUENCY                                **********  00012120
C*************************************************************        00012130
C                                                                       00012140
C                                                                       00012150
      DOUBLE PRECISION X,F,FCT,XLI,XRI,XL,XR,FL,FR,TOL,TOLF,A,DX,XM,FM  00012160
      DOUBLE PRECISION DSIGN,DABS                                       00012170
C                                                                       00012180
C     PREPARE ITERATION                                                 00012190
      IER=0                                                             00012200
      XL=XLI                                                            00012210
      XR=XRI                                                            00012220
      X=XL                                                              00012230
      TOL=X                                                             00012240
      F=FCT(TOL)                                                        00012250
      IF(F)1,16,1                                                       00012260
    1 FL=F                                                              00012270
      X=XR                                                              00012280
      TOL=X                                                             00012290
      F=FCT(TOL)                                                        00012300
      IF(F)2,16,2                                                       00012310
    2 FR=F                                                              00012320
      IF(DSIGN(1.D0,FL)+DSIGN(1.D0,FR))25,3,25                          00012330
C                                                                       00012340
C     BASIC ASSUMPTION FL*FR LESS THAN 0 IS SATISFIED.                  00012350
C     GENERATE TOLERANCE FOR FUNCTION VALUES.                           00012360
```

```
    3 I=0                                                                  00012370
      TOLF=100.*EPS                                                        00012380
C                                                                          00012390
CC                                                                         00012400
C     START ITERATION LOOP                                                 00012410
    4 I=I+1                                                                00012420
C                                                                          00012430
C     START BISECTION LOOP                                                 00012440
      DO 13 K=1,IEND                                                       00012450
      X=.5D0*(XL+XR)                                                       00012460
      TOL=X                                                                00012470
      F=FCT(TOL)                                                           00012480
      IF(F)5,16,5                                                          00012490
    5 IF(DSIGN(1.D0,F)+DSIGN(1.D0,FR))7,6,7                                00012500
C                                                                          00012510
C     INTERCHANGE XL AND XR IN ORDER TO GET THE SAME SIGN IN F AND FR      00012520
    6 TOL=XL                                                               00012530
      XL=XR                                                                00012540
      XR=TOL                                                               00012550
      TOL=FL                                                               00012560
      FL=FR                                                                00012570
      FR=TOL                                                               00012580
    7 TOL=F-FL                                                             00012590
      A=F*TOL                                                              00012600
      A=A+A                                                                00012610
      IF(A-FR*(FR-FL))8,9,9                                                00012620
    8 IF(I-IEND)17,17,9                                                    00012630
    9 XR=X                                                                 00012640
      FR=F                                                                 00012650
C                                                                          00012660
C     TEST ON SATISFACTORY ACCURACY IN BISECTION LOOP                       00012670
      TOL=EPS                                                              00012680
      A=DABS(XR)                                                           00012690
      IF(A-1.D0)11,11,10                                                   00012700
   10 TOL=TOL*A                                                            00012710
   11 IF(DABS(XR-XL)-TOL)12,12,13                                          00012720
   12 IF(DABS(FR-FL)-TOLF)14,14,13                                         00012730
   13 CONTINUE                                                             00012740
C     END OF BISECTION LOOP                                                00012750
CC                                                                         00012760
C     NO CONVERGENCE AFTER IEND ITERATION STEPS FOLLOWED BY IEND           00012770
C     SUCCESSIVE STEPS OF BISECTION OR STEADILY INCREASING FUNCTION        00012780
C     VALUES AT RIGHT BOUNDS. ERROR RETURN.                                00012790
      IER=1                                                                00012800
   14 IF(DABS(FR)-DABS(FL))16,16,15                                        00012810
   15 X=XL                                                                 00012820
      F=FL                                                                 00012830
   16 RETURN                                                               00012840
```

```
C                                                                       00012850
C     COMPUTATION OF ITERATED X-VALUE BY INVERSE PARABOLIC INTERPOLATION00012860
   17 A=FR-F                                                            00012870
      DX=(X-XL)*FL*(1.D0+F*(A-TOL)/(A*(FR-FL)))/TOL                     00012880
      XM=X                                                              00012890
      FM=F                                                              00012900
      X=XL-DX                                                           00012910
      TOL=X                                                             00012920
      F=FCT(TOL)                                                        00012930
      IF(F)18,16,18                                                     00012940
C                                                                       00012950
C     TEST ON SATISFACTORY ACCURACY IN ITERATION LOOP                   00012960
   18 TOL=EPS                                                           00012970
      A=DABS(X)                                                         00012980
      IF(A-1.D0)20,20,19                                                00012990
   19 TOL=TOL*A                                                         00013000
   20 IF(DABS(DX)-TOL)21,21,22                                          00013010
   21 IF(DABS(F)-TOLF)16,16,22                                          00013020
C                                                                       00013030
C     PREPARATION OF NEXT BISECTION LOOP                                00013040
   22 IF(DSIGN(1.D0,F)+DSIGN(1.D0,FL))24,23,24                          00013050
   23 XR=X                                                              00013060
      FR=F                                                              00013070
      GO TO 4                                                           00013080
   24 XL=X                                                              00013090
      FL=F                                                              00013100
      XR=XM                                                             00013110
      FR=FM                                                             00013120
      GO TO 4                                                           00013130
C     END OF ITERATION LOOP                                             00013140
C                                                                       00013150
C                                                                       00013160
C     ERROR RETURN IN CASE OF WRONG INPUT DATA                          00013170
   25 IER=2                                                             00013180
      RETURN                                                            00013190
      END                                                               00013200


      SUBROUTINE INVERT(A,N,D)                                          00013210
C*************************************************************        00013220
C********          SUBROUTINE INVERT                          **********00013230
C********     INVERTS THE 'R' MATRIX FOR                      **********00013240
C********        USE IN COMPUTING THE BRANCH                  **********00013250
C********        POINT MATRIX                                 **********00013260
C*************************************************************        00013270
      IMPLICIT REAL*8(A-H,O-Z)                                          00013280
      DIMENSION A(N,N),L(9),M(9)                                        00013290
C        SEARCH FOR LARGEST ELEMENT                                     00013300
```

```
      D=1.0                                                              00013310
      DO 80 K=1,N                                                        00013320
      L(K)=K                                                             00013330
      M(K)=K                                                             00013340
      BIGA=A(K,K)                                                        00013350
      DO 20 I=K,N                                                        00013360
      DO 20 J=K,N                                                        00013370
   10 IF(DABS(BIGA)-DABS(A(I,J))) 15,20,20                               00013380
   15 BIGA=A(I,J)                                                        00013390
      L(K)=I                                                             00013400
      M(K)=J                                                             00013410
   20 CONTINUE                                                           00013420
C        INTERCHANGE ROWS                                                00013430
      J=L(K)                                                             00013440
      IF(L(K)-K) 35,35,25                                                00013450
   25 DO 30 I=1,N                                                        00013460
      HOLD=-A(K,I)                                                       00013470
      A(K,I)=A(J,I)                                                      00013480
   30 A(J,I) =HOLD                                                       00013490
C        INTERCHANGE COLUMNS                                             00013500
   35 I=M(K)                                                             00013510
      IF(M(K)-K) 45,45,38                                                00013520
   38 DO 40 J=1,N                                                        00013530
      HOLD=-A(J,K)                                                       00013540
      A(J,K)=A(J,I)                                                      00013550
   40 A(J,I) =HOLD                                                       00013560
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS         00013570
C        CONTAINED IN BIGA)                                              00013580
   45 DO 55 I=1,N                                                        00013590
      IF(I-K) 50,55,50                                                   00013600
   50 A(I,K)=A(I,K)/(-A(K,K))                                            00013610
   55 CONTINUE                                                           00013620
C        REDUCE MATRIX                                                   00013630
      DO 65 I=1,N                                                        00013640
      DO 65 J=1,N                                                        00013650
      IF(I-K) 60,65,60                                                   00013660
   60 IF(J-K) 62,65,62                                                   00013670
   62 A(I,J)=A(I,K)*A(K,J)+A(I,J)                                        00013680
   65 CONTINUE                                                           00013690
C        DIVIDE ROW BY PIVOT                                             00013700
      DO 75 J=1,N                                                        00013710
      IF(J-K) 70,75,70                                                   00013720
   70 A(K,J)=A(K,J)/A(K,K)                                               00013730
   75 CONTINUE                                                           00013740
C        PRODUCT OF PIVOTS                                               00013750
      D=D*A(K,K)                                                         00013760
C        REPLACE PIVOT BY RECIPROCAL                                     00013770
      A(K,K)=1.0D0/A(K,K)                                                00013780
```

```
   80 CONTINUE                                                          00013790
C        FINAL ROW AND COLUMN INTERCHANGE                               00013800
      K=N                                                               00013810
  100 K=(K-1)                                                           00013820
      IF(K) 150,150,105                                                 00013830
  105 I=L(K)                                                            00013840
      IF(I-K) 120,120,108                                               00013850
  108 DO 110 J=1,N                                                      00013860
      HOLD=A(J,K)                                                       00013870
      A(J,K)=-A(J,I)                                                    00013880
  110 A(J,I) =HOLD                                                      00013890
  120 J=M(K)                                                            00013900
      IF(J-K) 100,100,125                                               00013910
  125 DO 130 I=1,N                                                      00013920
      HOLD=A(K,I)                                                       00013930
      A(K,I)=-A(J,I)                                                    00013940
  130 A(J,I) =HOLD                                                      00013950
      GO TO 100                                                         00013960
  150 RETURN                                                            00013970
      END                                                               00013980

      SUBROUTINE MATMUL(A,B,C,L,M)                                      00013990
C*************************************************************         00014000
C********          SUBROUTINE MATMUL                        **********  00014010
C********    MULTIPLIES TWO MATRICES ONE OF                 **********  00014020
C********       WHICH IS SQUARE                             **********  00014030
C*************************************************************         00014040
      IMPLICIT REAL*8(A-H,O-Z)                                          00014050
      DIMENSION A(L,L),B(L,M),C(L,M)                                    00014060
      DO 10 J=1,M                                                       00014070
      DO 10 I=1,L                                                       00014080
      C(I,J)=0.0D0                                                      00014090
      DO 10 K=1,L                                                       00014100
   10 C(I,J)=C(I,J)+A(I,K)*B(K,J)                                       00014110
      RETURN                                                            00014120
      END                                                               00014130

      SUBROUTINE POINT(RHO,D1,T,GAM,R,W,UPT)                            00014140
C*************************************************************         00014150
C********          SUBROUTINE POINT                         **********  00014160
C********   CONSTRUCTS A POINT MATRIX FOR A                 **********  00014170
C********          CURVED PIPE                              **********  00014180
C*************************************************************         00014190
      IMPLICIT REAL*8(A-H,O-Z)                                          00014200
      DIMENSION UPT(12,12)                                              00014210
      GO=32.174                                                         00014220
```

```
      RHOM=RHO/G0                                                          00014230
      PI=3.141592653589793                                                 00014240
      G=GAM                                                                00014250
      G1=GAM/2.D0                                                          00014260
      G2=2.D0*GAM                                                          00014270
      D2=D1-(2.D0*T)                                                       00014280
      RBAR=(D1+D2)/4.D0                                                    00014290
      AREA=PI*(D1**2-D2**2)/4.D0                                           00014300
      RG=R*DSIN(G1)/G1                                                     00014310
      XBAR=RG*DSIN(G1)                                                     00014320
      YBAR=R-RG*DCOS(G1)                                                   00014330
      DM=(RHOM*R*GAM*AREA)/20736.D0                                        00014340
      A1=(RHOM*AREA*R*RBAR**2)/41472.D0                                    00014350
      A2=(2.D0*RHOM*AREA*R**3)/20736.D0                                    00014360
      DIX=A1*(1.5D0*G+DSIN(G2)/4.D0)+A2*(.75D0*G-DSIN(G)+DSIN(G2)/8.D0)    00014370
      DIY=A1*(1.5D0*G-DSIN(G2)/4.D0)+A2*(.25D0*G-DSIN(G2)/8.D0)            00014380
      DIZ=A1*G+A2*(G-DSIN(G))                                              00014390
    7 DO 10 I=1,12                                                         00014400
      DO 10 J=1,12                                                         00014410
   10 UPT(I,J)=0.0D0                                                       00014420
      DO 20 I=1,12                                                         00014430
   20 UPT(I,I)=1.D0                                                        00014440
      W2=W**2                                                              00014450
      UPT(7,1)=-DM*W2                                                      00014460
      UPT(7,6)=-DM*W2*YBAR                                                 00014470
      UPT(8,2)=UPT(7,1)                                                    00014480
      UPT(9,3)=UPT(7,1)                                                    00014490
      UPT(8,6)=-DM*W2*XBAR                                                 00014500
      UPT(9,4)=-UPT(7,6)                                                   00014510
      UPT(9,5)=-UPT(8,6)                                                   00014520
      UPT(10,4)=-DIX*W2                                                    00014530
      UPT(11,5)=-DIY*W2                                                    00014540
      UPT(12,6)=-DIZ*W2                                                    00014550
      RETURN                                                               00014560
      END                                                                  00014570

      SUBROUTINE PRPLOT(X,Y,N,XMIN,XMAX,YMIN,YMAX,MODCUR)                  00014580
C*************************************************************           00014590
C********          SUBROUTINE PRPLOT                  **********           00014600
C********    PLOTS A DOUBLE PRECISION FUNCTION        **********           00014610
C********   LENGTHWISE ON TWO PAGES OF OUTPUT         **********           00014620
C*************************************************************           00014630
      DIMENSION GRID(161, 81),YSCALE(5),XSCALE(17)                         00014640
      DIMENSION Y(1),X(1)                                                  00014650
      INTEGER*2 GRID,BLANK,DASH,DOT,SLINE,YCHAR(4)/1H.,1H+,1H*,1HX/        00014660
      DATA DOT,BLANK,DASH,SLINE/Z4B40,Z4040,Z6D40,Z4F40/                   00014670
      NDATA=N*2                                                            00014680
```

```
      KKZ=2                                                              00014690
C                                                                        00014700
CC    RY IS THE NUMBER OF LENGTHWISE LINES OF OUTPUT PLOT DESIRED        00014710
C                                                                        00014720
      RY = 160.                                                          00014730
      IRYP1 = IFIX(RY) +1                                                00014740
      IXSC = IRYP1/10 + 1                                                00014750
      IERR=0                                                             00014760
C                                                                        00014770
CC    GRID IS THE MATRIX USED TO PLOT THE POINTS                         00014780
CC                                                                       00014790
CC    CHECKING Y AND X POINTS AND PLOTTING THOSE OUT OF RANGE            00014800
CC    AT THE MARGIN                                                      00014810
C                                                                        00014820
      DO 30 I=1,NDATA,KKZ                                                00014830
      IF(Y(I)-YMAX) 205,205,220                                          00014840
  220 Y  (I)=YMAX                                                        00014850
      IERR=IERR+1                                                        00014860
      GOTO 210                                                           00014870
  205 IF(Y(I)-YMIN)203,210,210                                           00014880
  203 Y  (I)=YMIN                                                        00014890
      IERR=IERR+1                                                        00014900
  210 IF(X(I)-XMAX)215,215,212                                           00014910
  212 X  (I)=XMAX                                                        00014920
      IERR=IERR+1                                                        00014930
      GOTO 30                                                            00014940
  215 IF(X(I)-XMIN)217,  30,30                                           00014950
  217 X  (I)=XMIN                                                        00014960
      IERR=IERR+1                                                        00014970
C                                                                        00014980
   30 CONTINUE                                                           00014990
C                                                                        00015000
CC    PLOTTING Y AND X AYIS , IF NECESSARY                               00015010
C                                                                        00015020
      YRANGE=YMAX-YMIN                                                   00015030
      XRANGE=XMAX-XMIN                                                   00015040
      IF (XRANGE.NE.0.)  GO TO 298                                       00015050
      XMIN=YMIN                                                          00015060
      XMAX=YMAX                                                          00015070
      XRANGE=YRANGE                                                      00015080
      GO TO 299                                                          00015090
  298 IF (YRANGE.NE.0.)  GO TO 299                                       00015100
      YMIN=XMIN                                                          00015110
      YMAX=XMAX                                                          00015120
      YRANGE=XRANGE                                                      00015130
C                                                                        00015140
CC    BLANKING OUT MATRIX-(GRID)                                         00015150
C                                                                        00015160
```

```
  299 DO 300 I=1,IRYP1                                                  00015170
      DO 301 JJ=1,81                                                    00015180
  301 GRID(I,JJ)=BLANK                                                  00015190
  300 CONTINUE                                                          00015200
      XTEST=XMAX*XMIN                                                   00015210
      YTEST=YMAX*YMIN                                                   00015220
      IF(YTEST)1,222,222                                                00015230
  222 IF(XTEST)333,444,444                                              00015240
    1 IXAXIS=80. *(-YMIN)/YRANGE+1.5                                    00015250
      DO 40 I=1,IRYP1                                                   00015260
   40 GRID(I,IXAXIS)=SLINE                                              00015270
      GOTO 222                                                          00015280
  333 IYAXIS=RY *XMAX/XRANGE+1.5                                        00015290
      DO 60 I=1,81                                                      00015300
   60 GRID(IYAXIS,I)=DASH                                               00015310
C                                                                       00015320
CC    PLACING POINTS IN THEIR PROPER GRID POSITIONS                     00015330
C                                                                       00015340
  444 IF(MODCUR.EQ.0.OR.MODCUR.EQ.1)JSET=0                              00015350
      JSET=JSET+1                                                       00015360
      IF(JSET.GT.4) JSET=1                                              00015370
      DO 700 I=1,NDATA,KKZ                                              00015380
      IPTY=RY *(X(I)-XMIN)/XRANGE+1.5                                   00015390
      IPTX=80. *(Y(I)-YMIN)/YRANGE+1.5                                  00015400
      IF(IPTY.GT.IRYP1.OR.IPTX.GT.81) GO TO 70                          00015410
      IF(IPTY.LE.0.OR.IPTX.LE.0)GO TO 70                                00015420
      GRID(IPTY,IPTX) = YCHAR(JSET)                                     00015430
      GO TO 700                                                         00015440
   70 IERR=IERR+1                                                       00015450
  700 CONTINUE                                                          00015460
C                                                                       00015470
CC    COMPUTE PROPER SCALE NUMBERS                                      00015480
C                                                                       00015490
      IF(MODCUR.EQ.0.OR.MODCUR.EQ.1) GO TO 8000                         00015500
      IF(MODCUR.EQ.2) RETURN                                            00015510
      GO TO 922                                                         00015520
 8000 YINCR=YRANGE/4.                                                   00015530
      XINCR=XRANGE/FLOAT(IXSC-1)                                        00015540
      YSCALE(1)=YMAX                                                    00015550
      XSCALE(1)=XMIN                                                    00015560
      DO 80 I=2,5                                                       00015570
   80 YSCALE(I)=YSCALE(I-1)-YINCR                                       00015580
      DO 81 I=2,IXSC                                                    00015590
   81 XSCALE(I)=XSCALE(I-1)+XINCR                                       00015600
C                                                                       00015610
CC    OUTPUT SECTION WITH GRAPH                                         00015620
C                                                                       00015630
      IF(MODCUR.EQ.0.OR.MODCUR.EQ.3)GO TO 922                           00015640
```

```
      RETURN                                                                    00015650
   17 FORMAT(12X,                                                               00015660
     1         1P E10.3,4(10X,E10.3)/15X,2H**,8(10H+*********),3H+**)           00015670
  922 WRITE(6,17) YSCALE(5),YSCALE(4),YSCALE(3),YSCALE(2),YSCALE(1)             00015680
      II=1                                                                      00015690
      I=0                                                                       00015700
      DO 101 IK=1,IRYP1                                                         00015710
      IF(I)91,91,92                                                             00015720
   91 WRITE(6,18) XSCALE(II),(GRID(IK,IY),IY=1,81),XSCALE(II)                   00015730
   18 FORMAT(3X,1P                                                              00015740
     1          E10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                          00015750
      II=II+1                                                                   00015760
      GOTO 102                                                                  00015770
   92 WRITE(6,19) (GRID(IK,IY),IY=1,81)                                         00015780
   19 FORMAT(15X,1H*,1X,81A1,1X,1H*)                                            00015790
  102 I=I+1                                                                     00015800
      IF(I-10)101,103,103                                                       00015810
  103 I=0                                                                       00015820
  101 CONTINUE                                                                  00015830
      WRITE(6,22) YSCALE(5),YSCALE(4),YSCALE(3),YSCALE(2),YSCALE(1)             00015840
   22 FORMAT(15X,2H**,8(10H+*********),3H+**/  1P                               00015850
     1                                        12X,E10.3,4(10X,E10.3))          00015860
      IF(IERR) 1000,1000,1001                                                   00015870
 1001 WRITE(6,20) IERR                                                          00015880
   20 FORMAT(10X 'NUMBER OF POINTS OUT OF RANGE =' I4)                          00015890
 1000 RETURN                                                                    00015900
      END                                                                       00015910

      SUBROUTINE SORT(X,Y,M)                                                    00015920
C*************************************************************               00015930
C********          SUBROUTINE SORT                          **********        00015940
C********     SORTS THE ARRAY OF FREQUENCIES                ***********       00015950
C********     AND VALUES OF THE FREQUENCY DE-               ***********       00015960
C******** TERMINANT FOR PLOTTING ON THE PRINTER             ***********       00015970
C*************************************************************               00015980
      IMPLICIT REAL*8(A-H,O-Z)                                                  00015990
      DIMENSION X(M),Y(M)                                                       00016000
      MCONST=M-1                                                                00016010
   10 DO 30 I=1,MCONST                                                          00016020
      IF(X(I+1)-X(I))20,30,30                                                   00016030
   20 A=X(I)                                                                    00016040
      B=Y(I)                                                                    00016050
      X(I)=X(I+1)                                                               00016060
      Y(I)=Y(I+1)                                                               00016070
      X(I+1)=A                                                                  00016080
      Y(I+1)=B                                                                  00016090
   30 CONTINUE                                                                  00016100
```

```
      MCONST=MCONST-1                                                   00016110
      IF(MCONST.EQ.1)GO TO 40                                           00016120
      GO TO 10                                                          00016130
   40 RETURN                                                            00016140
      END                                                               00016150

      SUBROUTINE STATEM(L,S)                                            00016160
C*************************************************************        00016170
C********          SUBROUTINE STATEM                  **********      00016180
C********     CONSTRUCTS THE STATE MATRIX FOR         **********      00016190
C********     FIXED, FREE, OR PINNED B.C.'S           **********      00016200
C*************************************************************        00016210
      IMPLICIT REAL*8(A-H,O-Z)                                          00016220
      DIMENSION S(12,6)                                                 00016230
      DO 10 J=1,6                                                       00016240
      DO 10 I=1,12                                                      00016250
   10 S(I,J)=0.0D0                                                      00016260
      IF(L-1)14,20,22                                                   00016270
   14 DO 15 I=1,6                                                       00016280
   15 S(I,I)=1.0D0                                                      00016290
      GO TO 26                                                          00016300
   20 DO 21 I=1,6                                                       00016310
   21 S(I+6,I)=1.D0                                                     00016320
      GO TO 26                                                          00016330
   22 DO 24 I=1,6                                                       00016340
   24 S(I+3,I)=1.D0                                                     00016350
   26 RETURN                                                            00016360
      END                                                               00016370

      SUBROUTINE STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UST)                00016380
C*************************************************************        00016390
C********          SUBROUTINE STRMAT                  **********      00016400
C********     CONSTRUCTS A TRANSFER MATRIX FOR        **********      00016410
C********     A STRAIGHT SECTION OF PIPE              **********      00016420
C*************************************************************        00016430
      IMPLICIT REAL*8(A-H,O-Z)                                          00016440
      DIMENSION UST(12,12)                                              00016450
      GO=32.174D0                                                       00016460
      PI=3.1415926535897 93                                             00016470
      G=EY/(2.D0*(1.D0+PR))                                             00016480
      D2=D1-(2.D0*T)                                                    00016490
      RY=DSQRT(D1**2+D2**2)/4.0D0                                       00016500
      RX=DSQRT((D1**2+D2**2)/8.D0)                                      00016510
      DJ=(PI*(D1**4-D2**4))/64.0D0                                      00016520
      DJT=2.D0*DJ                                                       00016530
      AREA=(PI*(D1**2-D2**2))/4.0D0                                     00016540
```

```
      UU=(RHO*AREA)/(20736.D0*G0)                                          00016550
      GAM=DL*W*DSQRT(UU/(AREA*G))                                          00016560
      AL=DL*W*DSQRT(UU/(AREA*EY))                                          00016570
      A=(DL**2)/(EY*DJ)                                                    00016580
      B4=(UU*(W**2)*(DL**4))/(EY*DJ)                                       00016590
      IF(MSR.EQ.0) GO TO 10                                                00016600
      V=1.D0-2.D0*T/D1                                                     00016610
      SDF=((7.D0+14.D0*PR+8.D0*PR**2)+((2.D0*V**2)/(1.D0+V**2)**2)         00016620
     1*(10.D0+20.D0*PR+8.D0*PR**2))/(6.D0*(1.D0+PR)**2)                    00016630
      SIG=(UU*(W**2)*(DL**2))/(SDF*G*AREA)                                 00016640
      TAU=((DL**2)*UU*(RY**2)*(W**2))/(EY*DJ)                              00016650
      GO TO 20                                                             00016660
   10 SIG=0.0D0                                                            00016670
      TAU=0.0D0                                                            00016680
   20 XL1=DSQRT(DSQRT(B4+.25D0*((SIG-TAU)**2))-.5D0*(SIG+TAU))             00016690
      XL2=DSQRT(DSQRT(B4+.25D0*((SIG-TAU)**2))+.5D0*(SIG+TAU))             00016700
      CL=1.0D0/(XL1**2+XL2**2)                                             00016710
      C0=CL*(XL2**2*DCOSH(XL1)+XL1**2*DCOS(XL2))                           00016720
      C1=CL*((XL2**2/XL1)*DSINH(XL1)+(XL1**2/XL2)*DSIN(XL2))               00016730
      C2=CL*(DCOSH(XL1)-DCOS(XL2))                                         00016740
      C3=CL*((1.0D0/XL1)*DSINH(XL1)-(1.0D0/XL2)*DSIN(XL2))                 00016750
      DO 30 I=1,12                                                         00016760
      DO 30 J=1,12                                                         00016770
   30 UST(I,J)=0.0D0                                                       00016780
      UST(1,1)=DCOS(AL)                                                    00016790
      UST(1,7)=(DL*DSIN(AL))/(EY*AREA*AL)                                  00016800
      UST(2,2)=C0-(SIG*C2)                                                 00016810
      UST(2,6)=DL*(C1-(SIG+TAU)*C3)                                        00016820
      UST(2,8)=-((A*DL)/B4)*(-SIG*C1+(B4+SIG**2)*C3)                       00016830
      UST(2,12)=A*C2                                                       00016840
      UST(3,3)=UST(2,2)                                                    00016850
      UST(3,5)=-UST(2,6)                                                   00016860
      UST(3,9)=UST(2,8)                                                    00016870
      UST(3,11)=-UST(2,12)                                                 00016880
      UST(4,4)=DCOS(GAM)                                                   00016890
      UST(4,10)=(DL*DSIN(GAM))/(G*DJT*GAM)                                 00016900
      UST(5,3)=-(B4*C3)/DL                                                 00016910
      UST(5,5)=C0-(TAU*C2)                                                 00016920
      UST(5,9)=UST(2,12)                                                   00016930
      UST(5,11)=(A/DL)*(C1-TAU*C3)                                         00016940
      UST(6,2)=-UST(5,3)                                                   00016950
      UST(6,6)=UST(5,5)                                                    00016960
      UST(6,8)=-A*C2                                                       00016970
      UST(6,12)=UST(5,11)                                                  00016980
      UST(7,1)=-(UU*DL*W**2*DSIN(AL))/AL                                   00016990
      UST(7,7)=UST(1,1)                                                    00017000
      UST(8,2)=-(B4*(C1-SIG*C3))/(A*DL)                                    00017010
      UST(8,6)=-(B4*C2)/A                                                  00017020
```

```
      UST(8,8)=UST(2,2)                                                 00017030
      UST(8,12)=UST(5,3)                                                00017040
      UST(9,3)=UST(8,2)                                                 00017050
      UST(9,5)=-UST(8,6)                                                00017060
      UST(9,9)=UST(2,2)                                                 00017070
      UST(9,11)=UST(6,2)                                                00017080
      UST(10,4)=-(DL*UU*RX**2*W**2*DSIN(GAM))/GAM                       00017090
      UST(10,10)=DCOS(GAM)                                              00017100
      UST(11,3)=UST(8,6)                                                00017110
      UST(11,5)=(DL/A)*(-TAU*C1+(B4+TAU**2)*C3)                         00017120
      UST(11,9)=UST(2,6)                                                00017130
      UST(11,11)=UST(5,5)                                               00017140
      UST(12,2)=UST(9,5)                                                00017150
      UST(12,6)=UST(11,5)                                               00017160
      UST(12,8)=-UST(11,9)                                              00017170
      UST(12,12)=UST(11,11)                                             00017180
      IF(MST.EQ.0)GO TO 41                                              00017190
      UST(3,3)=0.D0                                                     00017200
      UST(3,5)=1.D0                                                     00017210
      UST(3,9)=0.D0                                                     00017220
      UST(3,11)=1.D0                                                    00017230
      UST(5,3)=1.D0                                                     00017240
      UST(5,5)=0.D0                                                     00017250
      UST(5,9)=1.D0                                                     00017260
      UST(5,11)=0.D0                                                    00017270
      UST(9,3)=0.D0                                                     00017280
      UST(9,5)=1.D0                                                     00017290
      UST(9,9)=0.D0                                                     00017300
      UST(9,11)=1.D0                                                    00017310
      UST(11,3)=1.D0                                                    00017320
      UST(11,5)=0.D0                                                    00017330
      UST(11,9)=1.D0                                                    00017340
      UST(11,11)=0.D0                                                   00017350
   41 RETURN                                                            00017360
      END                                                               00017370

      SUBROUTINE UVEC(A,B,UX,UY,UZ,GAM)                                 00017380
C*************************************************************        00017390
C********          SUBROUTINE UVEC                          **********  00017400
C********   COMPUTES A SET OF ORTHOGONAL UNIT               **********  00017410
C******** VECTORS FROM TWO PIPE SECTION VECTORS             **********  00017420
C*************************************************************        00017440
      IMPLICIT REAL*8(A-H,O-Z)                                          00017450
      DIMENSION A(3),B(3),C(3),UX(3),UY(3),UZ(3)                        00017460
      AABS=DSQRT(A(1)**2+A(2)**2+A(3)**2)                               00017470
      DO 10 N=1,3                                                       00017480
   10 UX(N)=A(N)/AABS                                                   00017490
```

```
      C(1)=(B(2)*A(3)-A(2)*B(3))                                        00017500
      C(2)=(B(3)*A(1)-A(3)*B(1))                                        00017510
      C(3)=(B(1)*A(2)-A(1)*B(2))                                        00017520
      CABS=DSQRT(C(1)**2+C(2)**2+C(3)**2)                               00017530
      DO 20 N=1,3                                                       00017540
   20 UZ(N)=C(N)/CABS                                                   00017550
      UY(1)=(UZ(2)*UX(3)-UX(2)*UZ(3))                                   00017560
      UY(2)=(UZ(3)*UX(1)-UX(3)*UZ(1))                                   00017570
      UY(3)=(UZ(1)*UX(2)-UX(1)*UZ(2))                                   00017580
      BABS=DSQRT(B(1)**2+B(2)**2+B(3)**2)                               00017590
      BDOTA=B(1)*A(1)+B(2)*A(2)+B(3)*A(3)                               00017600
      GAM=DARCOS(BDOTA/(BABS*AABS))                                     00017610
      RETURN                                                            00017620
      END                                                               00017630

      SUBROUTINE WSTART(W,DL,RHO,EY,PR,D1,T,MSR,MST,D)                  00017640
C*****************************************************************     00017650
C*********          SUBROUTINE WSTART                  ***********     00017660
C*********    COMPUTES THE VALUE OF THE                ***********     00017670
C*********    FREQUENCY DETERMINANT FOR A SYNTHETIC    ***********     00017680
C*********    STRAIGHT PIPE EQUAL IN LENGTH TO         ***********     00017690
C*********    THE MAIN MEMBER AND HAVING WEIGHTED      ***********     00017700
C*********    AVERAGE VALUES OF THE PROPERTIES OF      ***********     00017710
C*********    THE MAIN MEMBER SECTIONS                 ***********     00017720
C*****************************************************************     00017730
      IMPLICIT REAL*8(A-H,O-Z)                                          00017740
      DIMENSION UST(12,12),DD(6,6)                                      00017750
    6 CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UST)                      00017760
    7 DO 10 I=1,6                                                       00017770
    8 DO 10 J=1,6                                                       00017780
   10 DD(I,J)=UST(I,J+6)                                                00017790
   11 CALL DETER(6,DD,D)                                                00017800
   12 CONTINUE                                                          00017810
      RETURN                                                            00017820
      END                                                               00017830

      FUNCTION FRDET(W)                                                 00017840
C*****************************************************************     00017850
C*********          FUNCTION FRDET                     ***********     00017860
C*********    COMPUTES THE TRANSFER MATRIX FOR         ***********     00017870
C*********    EACH SECTION OF PIPE AND EACH            ***********     00017880
C*********    COORDINATE TRANSFORMATION, MUL-          ***********     00017890
C*********    TIPLIES THEM TOGETHER AND COMPUTES       ***********     00017900
C*********    THE FREQUENCY DETERMINANT                ***********     00017910
C*****************************************************************     00017920
      IMPLICIT REAL*8(A-H,O-Z)                                          00017930
```

```
      COMMON/VIB1/AA(100,10)                                                    00017940
      COMMON/VIB2/NB(50)                                                        00017950
      COMMON/VIB3/MS(2)                                                         00017960
      DIMENSION U(12,6),U1(12,6),U2(12,6),UA(12,12),UB(12,12),U3(12,6),         00017970
     1CTX12(12,12),CTZ12(12,12),PT(12,12),R1(6,6),R2(6,6),G(6,6),G1(6,6)00017980
     2,G2(6,6),G3(6,6),G3T(6,6),G2T(6,6),G1T(6,6),DD(6,6)                       00017990
      MSR=MS(1)                                                                 00018000
      MST=MS(2)                                                                 00018010
      LBC=NB(1)                                                                 00018020
      N=1                                                                       00018030
      K=3                                                                       00018040
C                                                                               00018050
C     SET UP INITIAL MAIN MEMBER STATE MATRIX                                   00018060
C                                                                               00018070
      CALL STATEM(LBC,U)                                                        00018080
  100 ISEC=AA(N,1)                                                              00018090
      GO TO(110,200,300,500,400,400,430),ISEC                                   00018100
C                                                                               00018110
C     FORM STRAIGHT SECTION TRANSFER MATRIX AND MULTIPLY TIMES SYSTEM           00018120
C     TRANSFER MATRIX                                                           00018130
C                                                                               00018140
  110 EY=AA(N,2)                                                                00018150
      PR=AA(N,3)                                                                00018160
      RHO=AA(N,4)                                                               00018170
      D1=AA(N,5)                                                                00018180
      T=AA(N,6)                                                                 00018190
      DL=AA(N,7)                                                                00018200
      CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)                               00018210
      CALL MATMUL(UA,U,U1,12,6)                                                 00018220
      DO 120 I=1,12                                                             00018230
      DO 120 J=1,6                                                              00018240
  120 U(I,J)=U1(I,J)                                                            00018250
      N=N+1                                                                     00018260
      GO TO 100                                                                 00018270
C                                                                               00018280
C     FORM POINT,FIELD,AND COORDINATE TRANSFORMATION MATRICES FOR  CURVED       00018290
C       SECTION AND MULTIPLY TIMES THE SYSTEM TRANSFER MATRIX                   00018300
C                                                                               00018310
  200 EY=AA(N,2)                                                                00018320
      PR=AA(N,3)                                                                00018330
      RHO=AA(N,4)                                                               00018340
      D1=AA(N,5)                                                                00018350
      T=AA(N,6)                                                                 00018360
      R=AA(N,7)                                                                 00018370
      ALPHA=AA(N,8)                                                             00018380
      GAM=AA(N,9)                                                               00018390
      NSS=AA(N,10)                                                              00018400
  203 IF(ALPHA.GT.0.01D0)GO TO 206                                              00018410
```

```
      DO 204 I=1,12                                                    00018420
      DO 204 J=1,12                                                    00018430
  204 U1(I,J)=U(I,J)                                                   00018440
      GO TO 210                                                        00018450
  206 CALL COORDX(ALPHA,12,CTX12)                                      00018460
      CALL MATMUL(CTX12,U,U1,12,6)                                     00018470
  210 GAMS=GAM/DFLOAT(NSS)                                             00018480
  212 CALL POINT(RHO,D1,T,GAMS,R,W,UA)                                 00018490
      CALL CURMAT(EY,PR,D1,R,T,GAMS,MSR,UB)                            00018500
  220 CALL MATMUL(UA,U1,U2,12,6)                                       00018510
      CALL MATMUL(UB,U2,U1,12,6)                                       00018520
      NSS=NSS-1                                                        00018530
  223 IF(NSS.GT.0)GO TO 220                                            00018540
      DO 230 I=1,12                                                    00018550
      DO 230 J=1,6                                                     00018560
  230 U(I,J)=U1(I,J)                                                   00018570
      N=N+1                                                            00018580
      GO TO 100                                                        00018590
C                                                                      00018600
CC    FORM TRANSFER MATRICES FOR CORNER SECTIONS AND MULTIPLY TIMES     00018610
C       SYSTEM TRANSFER MATRIX                                         00018620
C                                                                      00018630
  300 EY=AA(N,2)                                                       00018640
      PR=AA(N,3)                                                       00018650
      RHO=AA(N,4)                                                      00018660
      D1=AA(N,5)                                                       00018670
      T=AA(N,6)                                                        00018680
      DL=AA(N,7)                                                       00018690
      ALPHA=AA(N,8)                                                    00018700
      GAM=AA(N,9)                                                      00018710
      IF(ALPHA.GT.0.01D0)GO TO 306                                     00018720
      DO 304 I=1,12                                                    00018730
      DO 304 J=1,12                                                    00018740
  304 U1(I,J)=U(I,J)                                                   00018750
      GO TO 310                                                        00018760
  306 CALL COORDX(ALPHA,12,CTX12)                                      00018770
      CALL MATMUL(CTX12,U,U1,12,6)                                     00018780
  310 CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)                      00018790
      CALL MATMUL(UA,U1,U2,12,6)                                       00018800
      CALL COORDZ(GAM,12,CTZ12)                                        00018810
      CALL MATMUL(CTZ12,U2,U,12,6)                                     00018820
      N=N+1                                                            00018830
      GO TO 100                                                        00018840
C                                                                      00018850
CC    FORM TRANSFER MATRIX FOR THE STRAIGHT SECTION PRECEEDING A       00018860
C       BRANCH AND MULTIPLY TIMES THE SYSTEM TRANSFER MATRIX           00018870
C                                                                      00018880
  400 NK=0                                                             00018890
```

```
      IF(ISEC.EQ.5)GO TO 410                                              00018900
      NK=1                                                                00018910
  410 EY=AA(N,2)                                                          00018920
      PR=AA(N,3)                                                          00018930
      RHO=AA(N,4)                                                         00018940
      D1=AA(N,5)                                                          00018950
      T=AA(N,6)                                                           00018960
      DL=AA(N,7)                                                          00018970
      CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)                         00018980
      CALL MATMUL(UA,U,U3,12,6)                                           00018990
      N=N+1                                                               00019000
  420 JBC=NB(K)                                                           00019010
      K=K+1                                                               00019020
C                                                                         00019030
CC    SET UP INITIAL STATE MATRIX FOR THE END OF THE BRANCH               00019040
C                                                                         00019050
      CALL STATEM(JBC,U)                                                  00019060
C                                                                         00019070
CC    USE MAIN MEMBER ROUTINES TO COMPUTE THE BRANCH TRANSFER MATRIX      00019080
C                                                                         00019090
      GO TO 100                                                           00019100
  430 EY=AA(N,2)                                                          00019110
      PR=AA(N,3)                                                          00019120
      RHO=AA(N,4)                                                         00019130
      D1=AA(N,5)                                                          00019140
      T=AA(N,6)                                                           00019150
      DL=AA(N,7)                                                          00019160
      ALPHA1=AA(N,8)                                                      00019170
      GAM=AA(N,9)                                                         00019180
      ALPHA=AA(N,10)                                                      00019190
      CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)                         00019200
      CALL MATMUL(UA,U,U1,12,6)                                           00019210
C                                                                         00019220
CC    COMPUTE THE BRANCH POINT MATRIX AND MULTIPLY TIMES THE SYSTEM       00019230
CCC     TRANSFER MATRIX                                                   00019240
C                                                                         00019250
      DO 440 I=1,6                                                        00019260
      DO 440 J=1,6                                                        00019270
      R1(I,J)=U1(I,J)                                                     00019280
  440 R2(I,J)=U1(I+6,J)                                                   00019290
  442 CALL INVERT(R1,6,DET)                                               00019300
      CALL COORDX(ALPHA,6,G2)                                             00019310
      CALL COORDX(ALPHA1,6,G1)                                            00019320
      CALL COORDZ(GAM,6,G3)                                               00019330
      DO 450 I=1,6                                                        00019340
      DO 450 J=1,6                                                        00019350
      G3T(I,J)=G3(J,I)                                                    00019360
      G2T(I,J)=G2(J,I)                                                    00019370
```

```
  450 G1T(I,J)=G1(J,I)                                                    00019380
      CALL MATMUL(G3T,G2T,G,6,6)                                          00019390
      CALL MATMUL(G1T,G,G2T,6,6)                                          00019400
      CALL MATMUL(R1,G2T,G3T,6,6)                                         00019410
      CALL MATMUL(R2,G3T,G2T,6,6)                                         00019420
      CALL MATMUL(G1,G2T,G3T,6,6)                                         00019430
      CALL MATMUL(G3,G3T,G2T,6,6)                                         00019440
  457 CALL MATMUL(G2,G2T,G,6,6)                                           00019450
      DO 460 I=1,12                                                       00019460
      DO 460 J=1,12                                                       00019470
  460 PT(I,J)=0.D0                                                        00019480
      DO 465 I=1,12                                                       00019490
  465 PT(I,I)=1.D0                                                        00019500
      DO 470 I=1,6                                                        00019510
      DO 470 J=1,6                                                        00019520
  470 PT(I+6,J)=G(I,J)                                                    00019530
  471 CALL MATMUL(PT,U3,U,12,6)                                           00019540
C                                                                         00019550
CC    CHECK TO SEE IF JUNCTION CONTAINS A DUAL-BRANCH                     00019560
C                                                                         00019570
      N=N+1                                                               00019580
      IF(NK.EQ.0)GO TO 100                                                00019590
      NK=NK-1                                                             00019600
      DO 480 I=1,12                                                       00019610
      DO 480 J=1,6                                                        00019620
  480 U3(I,J)=U(I,J)                                                      00019630
      GO TO 420                                                           00019640
C                                                                         00019650
CC    FORM TRANSFER MATRIX FOR END SECTION(IF IF LENGTH IS NONZERO) AND    00019660
CC      MULTIPLY TIMES SYSTEM TRANSFER MATRIX                             00019670
C                                                                         00019680
  500 EY=AA(N,2)                                                          00019690
      PR=AA(N,3)                                                          00019700
      RHO=AA(N,4)                                                         00019710
      D1=AA(N,5)                                                          00019720
      T=AA(N,6)                                                           00019730
      DL=AA(N,7)                                                          00019740
      W1=W                                                                00019750
      IF(DL.GT.0.D0)GO TO 510                                             00019760
      DO 505 I=1,6                                                        00019770
      DO 505 J=1,6                                                        00019780
  505 U1(I,J)=U(I,J)                                                      00019790
  506 GO TO 520                                                           00019800
  510 CALL STRMAT(W,DL,RHO,EY,PR,D1,T,MSR,MST,UA)                         00019810
      CALL MATMUL(UA,U,U1,12,6)                                           00019820
C                                                                         00019830
CC    SET UP FREQUENCY DETERMINANT ACCORDING TO RIGHT END BOUNDARY CON-   00019840
C       DITION                                                            00019850
```

```
C                                                                       00019860
  520 MBC=NB(2)                                                         00019870
      IF(MBC-1)522,530,536                                              00019880
  522 DO 525 I=1,6                                                      00019890
      DO 525 J=1,6                                                      00019900
  525 DD(I,J)=U1(I+6,J)                                                 00019910
      GO TO 540                                                         00019920
  530 DO 535 I=1,6                                                      00019930
      DO 535 J=1,6                                                      00019940
  535 DD(I,J)=U1(I,J)                                                   00019950
      GO TO 540                                                         00019960
  536 DO 537 I=1,3                                                      00019970
      DO 537 J=1,6                                                      00019980
  537 DD(I,J)=U1(I,J)                                                   00019990
      DO 538 I=4,6                                                      00020000
      DO 538 J=1,6                                                      00020010
  538 DD(I,J)=U1(I+6,J)                                                 00020020
C                                                                       00020030
C     COMPUTE FREQUENCY DETERMINANT                                     00020040
C                                                                       00020050
  540 CALL DETER(6,DD,D)                                                00020060
      FRDET=D                                                           00020070
  560 RETURN                                                            00020080
      END                                                               00020090
```

APPENDIX F
PROGRAM ACCURACY AND INTEGRITY ANALYSIS

1. General Remarks

This appendix tabulates by mode frequency (radians/second) the results of the accuracy and integrity analyses of various piping systems from simple to complex.

The comparison frequency values for the simple straight section systems were obtained by solution of the differential equations for longitudinal, torsional and flexural vibrations with application of the appropriate boundary conditions. These solutions were evaluated on the digital computer with accuracy to at least as many places as listed in the tabulation. Evaluation of the comparison solution and VIBREL solution were carried out using a value of 32.174 FT/SEC^2 for the gravitational constant and a value of π to 15 significant figures.

Comparison values for the curved pipe analysis were obtained from three sources: Fink [5], Kim [6], and Ojalvo [8]. Those extracted directly from [5] and [6] were compiled using a lumped mass model and digital computation. From graphs and formulas (based on the theory of clamped ring segments) appearing in [8], the other natural frequency values for the curved pipe were calculated.

Several comparison frequencies for a single branch system were obtained from [6].

Shear deflection and rotary inertia were neglected except where noted.

2. SYSTEMS ANALYSIS

a. STRAIGHT PIPE WITH VARIOUS END CONDITIONS

END CONDITIONS:

FIXED-FIXED

SYSTEM PARAMETERS:

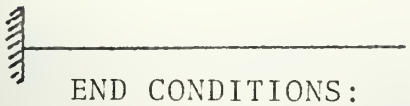
DIAMETER (IN)	2.0
WALL THICKNESS (IN)	.125
LENGTH (IN)	200.
MODULUS OF ELASTICITY (PSI)	30×10^6
POISSON'S RATIO	.25
WEIGHT DENSITY (LB/FT^3)	490.

COMPARISON RESULTS:

MODE	ANALYTICAL	VIBREL	% DIFFERENCE
1ST FLEXURAL	75.10461420	75.10461590	.00000
2ND FLEXURAL	207.02876110	207.02876110	.00000
3RD FLEXURAL	405.85914940	405.85914210	.00000
4TH FLEXURAL	670.90578980	670.90577600	.00000
5TH FLEXUPAL	1002.21750380	1002.21750410	.00000
6TH FLEXURAL	1399.79137910	1399.79140510	.00000
7TH FLEXURAL	1863.62757630	1863.62799710	.00002
1ST TORSIONAL	2007.8325054	2007.8275305	.00025
8TH FLEXURAL	2393.7260868	2393.7157896	.00043
9TH FLEXURAL	2990.0869112	2990.3653337	.00931
1ST LONGITUDINAL	3174.6619386	3174.6560265	.00019

DUAL ANALYSIS RESULTS:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	75.1046142	75.1046142	.00000
2	207.0287611	207.0287611	.00000
3	405.8591494	405.8591494	.00000
4	670.9057898	670.9057898	.00000
5	1002.2175038	1002.2175038	.00000
6	1399.7913791	1399.7913791	.00000
7	1863.6275763	1863.6275763	.00000



END CONDITIONS:
FIXED-FREE

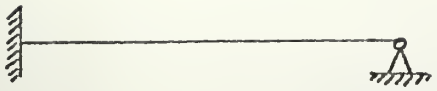
SYSTEM PARAMETERS:
SAME AS (1)

COMPARISON RESULTS:

MODE	ANALYTICAL	VIBREL	% DIFFERENCE
1	11.8028657	11.8028696	.00003
2	73.9673211	73.9673092	.00002
3	207.1106480	207.1106397	.00000
4	405.8541957	405.8541953	.00000
5	670.9060651	670.9060651	.00000
6	1399.7913799	1399.7913885	.00000

DUAL ANALYSIS RESULTS:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	11.8028696	11.8028696	.00000
2	73.9673092	73.9673092	.00000
3	207.1106397	207.1106397	.00000
4	405.8541953	405.8541953	.00000
5	670.9060651	670.9060651	.00000
6	1399.7913885	1399.7913885	.00000



SYSTEM PARAMETERS:

SAME AS (1)

END CONDITIONS:

FIXED-PINNED

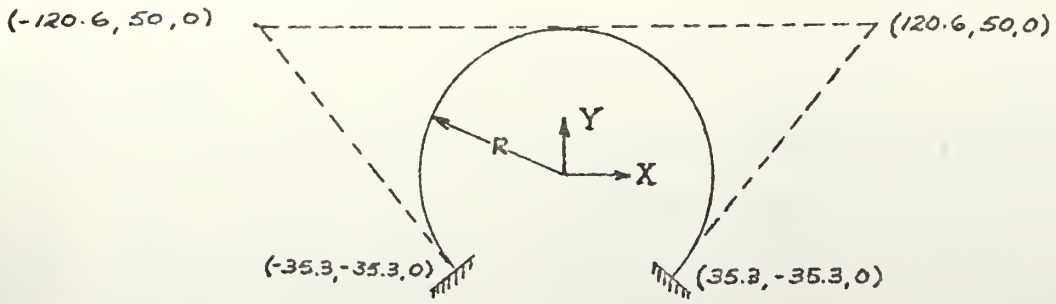
COMPARISON RESULTS:

MODE	ANALYTICAL	VIBREL	% DIFFERENCE
1	51.7571905	51.7571773	.00003
2	167.7264474	167.7264180	.00002
3	349.9478448	349.9478460	.00000
4	598.4315201	598.4315226	.00000
5	913.1775124	913.17753170	.00000
6	1003.9162520	1003.9164796	.00002

DUAL ANALYSIS RESULTS:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	51.7571773	51.7571773	.00000
2	167.7264180	167.7264180	.00000
3	349.9478460	349.9478460	.00000
4	598.4315226	598.4315226	.00000
5	913.17753170	913.17753170	.00000
6	1003.9164796	1003.9164796	.00000

b. CURVED PIPE



END CONDITIONS: FIXED-FIXED

SYSTEM PARAMETERS:

PIPE DIAMETER (IN)	2.0
WALL THICKNESS (IN)	.125
INCLUDED ANGLE OF ARC (DEG)	270
MODULUS OF ELASTICITY (PSI)	30×10^6
POISSON'S RATIO	.25
WEIGHT DENSITY (LB/FT^3)	556

COMPARISON RESULTS

a. MODE 1

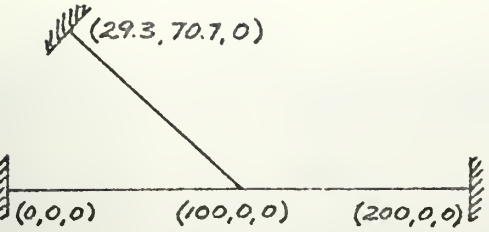
SOURCE	FREQUENCY	% VARIATION WITH VIBREL VALUE
VIBREL	37.2234058	
OJALVO [8]	38.5930983	3.68
FINK [5]	38.8773520	4.44
KIM [6]	35.3639740	-5.00

b. MODE 2

SOURCE	FREQUENCY	% VARIATION WITH VIBREL VALUE
VIBREL	72.1914615	
OJALVO	69.9962862	-3.04
FINK	73.3297307	1.58
KIM	66.6993490	-7.61

c. BRANCHED SYSTEMS

(1) SINGLE BRANCH SYSTEM



SYSTEM PARAMETERS (ALL 3 SECTIONS):

DIAMETER (IN)	2.0
WALL THICKNESS (IN)	.125
LENGTH (IN)	100.
MODULUS OF ELASTICITY (PSI)	30×10^6
POISSON'S RATIO	.25
WEIGHT DENSITY (LB/FT^3)	556.

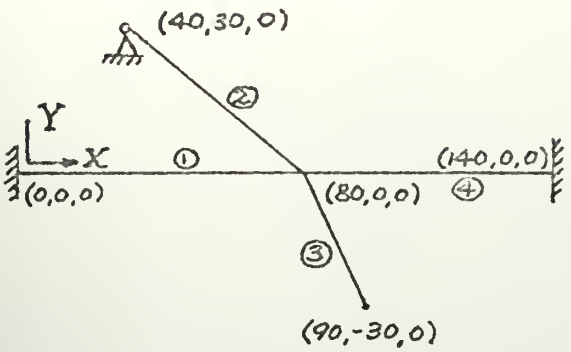
COMPARISON RESULTS:

MODE	KIM [6]	VIBREL	% DIFFERENCE
2	194.25470558	194.2888499	.01758
5	278.61078241	278.6390155	.01013
8	628.59556355	628.9417782	.05508

DUAL ANALYSIS RESULTS:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	67.6456627	67.6456627	.00000
2	194.2888474	194.2888474	.00000
3	198.0817333	198.0817333	.00000
4	242.0773169	242.0773169	.00000
5	278.6390155	278.6390155	.00000
6	281.9864461	281.9864461	.00000
7	370.7949928	370.7949928	.00000
8	628.9417782	628.9417782	.00000
9	634.7478785	634.7478786	.00000
10	689.7373111	689.7373111	.00000

(2) DUAL BRANCH SYSTEM



END CONDITIONS:

MAIN MEMBER - FIXED-FIXED
BRANCH #1 - PINNED
BRANCH #2 - FREE

SYSTEM PARAMETERS:

PARAMETER	SECTION 1	SECTION 2	SECTION 3	SECTION 4
DIAMETER (IN)	2.0	2.0	2.0	2.0
WALL THICKNESS (IN)	.125	.125	.125	.125
LENGTH (IN)	80.0	50.0	31.6	60.0
MOD. OF ELAST. (PSI)	30×10^6	30×10^6	30×10^6	30×10^6
POISSON'S RATIO	.30	.30	.30	.30
WT. DENSITY (LB/FT^3)	490.0	490.0	490.0	490.0

DUAL ANALYSIS RESULTS*:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	110.5015347	110.5015347	.00000
2	276.6928874	276.6928874	.00000
3	314.3597336	314.3597336	.00000
4	417.0897739	417.0897739	.00000
5	463.5288123	463.5288123	.00000
6	675.6526889	675.6526889	.00000

* SHEAR DEFLECTION AND ROTARY INERTIA ARE CONSIDERED IN THIS ANALYSIS.

† COORDINATES SHOWN IN DIAGRAM ARE IN INCHES.

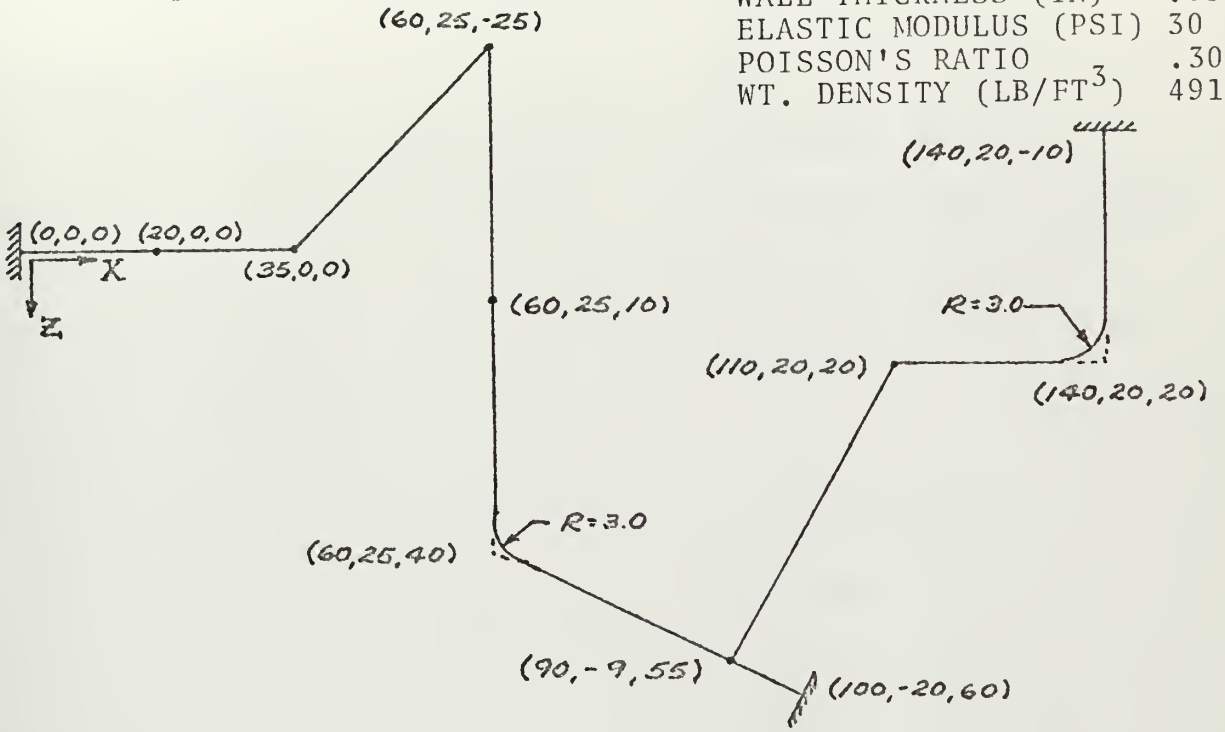
d. COMPLEX SYSTEM

END CONDITIONS:

MAIN MEMBER - FIXED-FIXED
BRANCH - FIXED

SYSTEM PARAMETERS
(ALL SECTIONS):

DIAMETER (IN)	2.25
WALL THICKNESS (IN)	.035
ELASTIC MODULUS (PSI)	30×10^6
POISSON'S RATIO	.30
WT. DENSITY (LB/FT^3)	491.5



DUAL ANALYSIS RESULTS:

MODE	STARTING AT LEFT END	STARTING AT RIGHT END	% DIFFERENCE
1	73.0350449	73.7545503	.985
2	102.7404907	102.9266533	.181
3	142.5065200	146.3168530	2.674
4	201.7233189	201.3564078	.182
5	239.1786608	239.1641100	.006
6	327.8210841	321.1763486	2.027
7	331.8165320	337.3328726	1.662
8	364.5803428	364.1833110	.109
9	428.5622381	429.1841008	.145
10	520.1734071	521.2514969	.207
11	542.4026997	544.3290021	.355
12	565.0762600	564.3679439	.125
13	574.0701684	571.0709491	.522
14	648.8240390	651.2831922	.379
15	754.8640271	759.4399716	.606
16	1052.0436077	1052.3129250	.026
17	1210.9202599	1210.7321375	.016
18	1250.1288524	1247.3298693	.224

APPENDIX G

SAMPLE PROBLEM

1. Piping System

The piping used in this example is identical to the complex system shown in Appendix F. It includes several corners and bends and has one branch junction point. The boundary conditions are fixed for the two ends of the main member and the remote branch end.

2. Data Cards

Shown in Fig. G.1 is the arrangement of data cards as they would appear following the program deck. Note that Card #1 specifies one system to be analyzed, and Card #2 asks for four mode frequencies in the analysis. Since the curved subsection override is not utilized, the program will go ahead and determine the number of curved pipe subdivisions according to the length-to-diameter ratio. Shear deflection and rotary inertia are not included in this problem analysis.

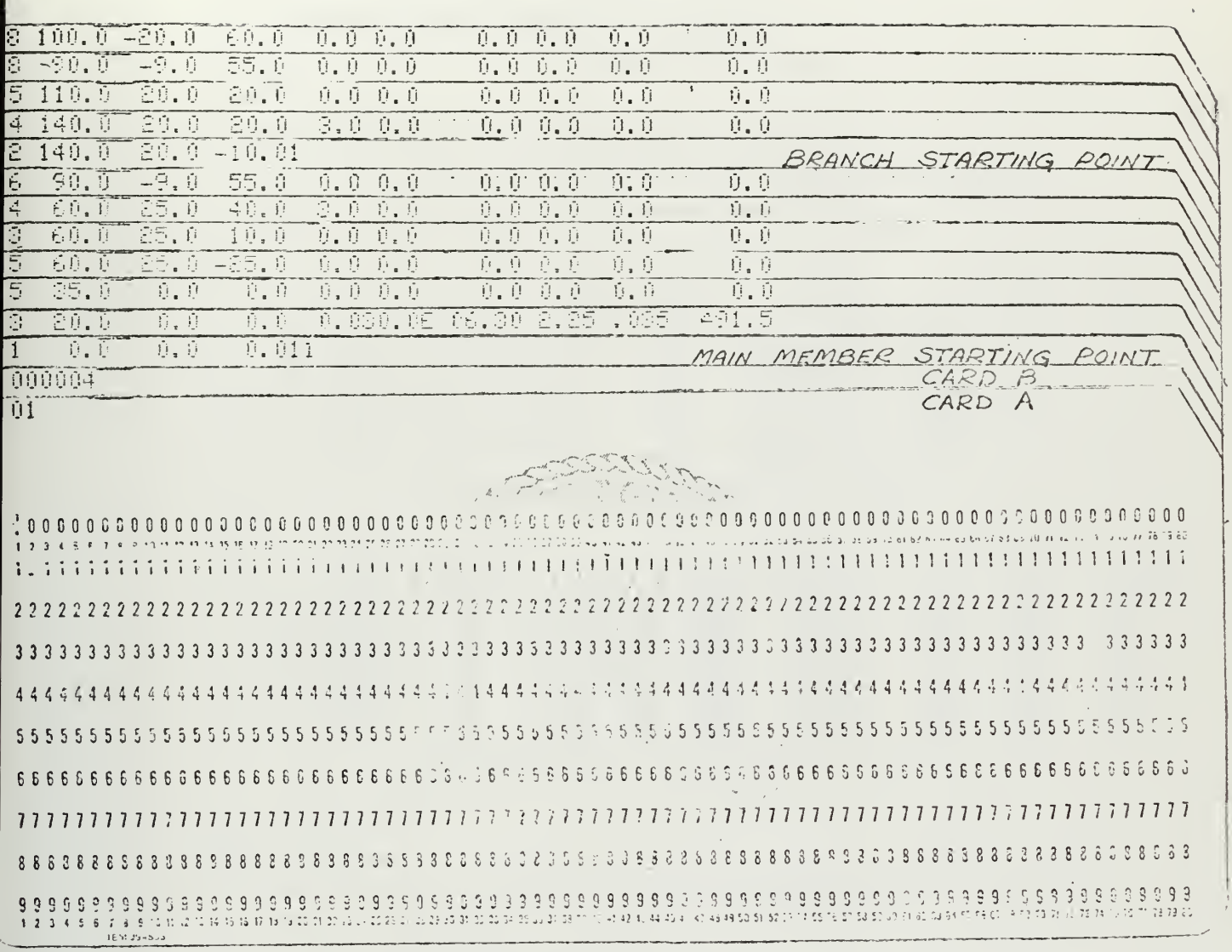


Figure G.1: Data Deck for Sample Program

```
**************************************************************************
*                                                                        *
*                PROGRAM  VIBREL                  C.O.RUOOLF             *
*                                                                        *
**************************************************************************
```

PROGRAM VIBREL COMPUTES THE NATURAL FREQUENCIES OF VIBRATION
OF ANY RANDOMLY ARRANGED 3-DIMENSIONAL PIPING SYSTEM
PIPING HANGERS AND PROJECTIVE COMPLEXITIES ARE NOT INCLUDED

****

PROBLEM # 1

****

THE EFFECTS OF SHEAR DEFLECTION AND ROTARY INERTIA ARE NOT CONSIDERED IN THIS PROBLEM

****

INPUT DATA:

ID	X COORD	Y COORD	Z COORD	RAD OF CURV	ELASTIC MODULUS	POISSONS RATIO	O.D.	WALL THICK	WT/VOL	LEFT BC CODE	RIGHT BC CODE
1	0.0	0.0	0.0							1	1
3	20.0	0.0	0.0	0.0	0.3000 0B	.30	2.25	0.0350	491.5		
5	35.0	0.0	0.0	0.0	0.0	.0	0.0	0.0	0.0		
5	60.0	25.0	-25.0	0.0	0.0	.0	0.0	0.0	0.0		
3	60.0	25.0	10.0	0.0	0.0	.0	0.0	0.0	0.0		
4	60.0	25.0	40.0	3.0	0.0	.0	0.0	0.0	0.0		
6	90.0	-9.0	55.0	0.0	0.0	.0	0.0	0.0	0.0		
2	140.0	20.0	-10.0							1	
4	140.0	20.0	20.0	3.0	0.0	.0	0.0	0.0	0.0		
5	110.0	20.0	20.0	0.0	0.0	.0	0.0	0.0	0.0		
B	90.0	-9.0	55.0	0.0	0.0	.0	0.0	0.0	0.0		
B	100.0	-20.0	60.0	0.0	0.0	.0	0.0	0.0	0.0		

****

PROPERTIES AND GEOMETRY:

SECT NR	ID	ELASTIC MODULUS	POISSONS RATIO	WT/VOL	O.D.	WALL THICK	LENGTH OR RAD OF CURV	ALPHA1	GAMMA	NR OF SUBSEC OR ALPHA
1	1.0	0.3000 0B	.30	491.5	2.25	0.0350	20.000	0.0	0.0	0.0
2	3.0	0.3000 08	.30	491.5	2.25	0.0350	15.000	-2.356	0.955	0.0
3	3.0	0.3000 08	.30	491.5	2.25	0.0350	43.301	2.094	2.186	0.0
4	1.0	0.3000 08	.30	491.5	2.25	0.0350	35.000	0.0	0.0	0.0
5	1.0	0.3000 0B	.30	491.5	2.25	0.0350	27.833	0.0	0.0	0.0
6	2.0	0.3000 08	.30	491.5	2.25	0.0350	3.000	1.508	1.251	2.000
7	5.0	0.3000 08	.30	491.5	2.25	0.0350	45.592	0.0	0.0	0.0
8	1.0	0.3000 08	.30	491.5	2.25	0.0350	27.000	0.0	0.0	0.0
9	2.0	0.3000 0B	.30	491.5	2.25	0.0350	3.000	0.0	1.571	2.000
10	3.0	0.3000 0B	.30	491.5	2.25	0.0350	27.000	2.450	1.156	0.0
11	7.0	0.3000 08	.30	491.5	2.25	0.0350	49.659	-0.372	-1.177	-0.847
12	4.0	0.3000 08	.30	491.5	2.25	0.0350	15.684	0.0	0.0	0.0

****

BOUNDARY CONDITION CODES:

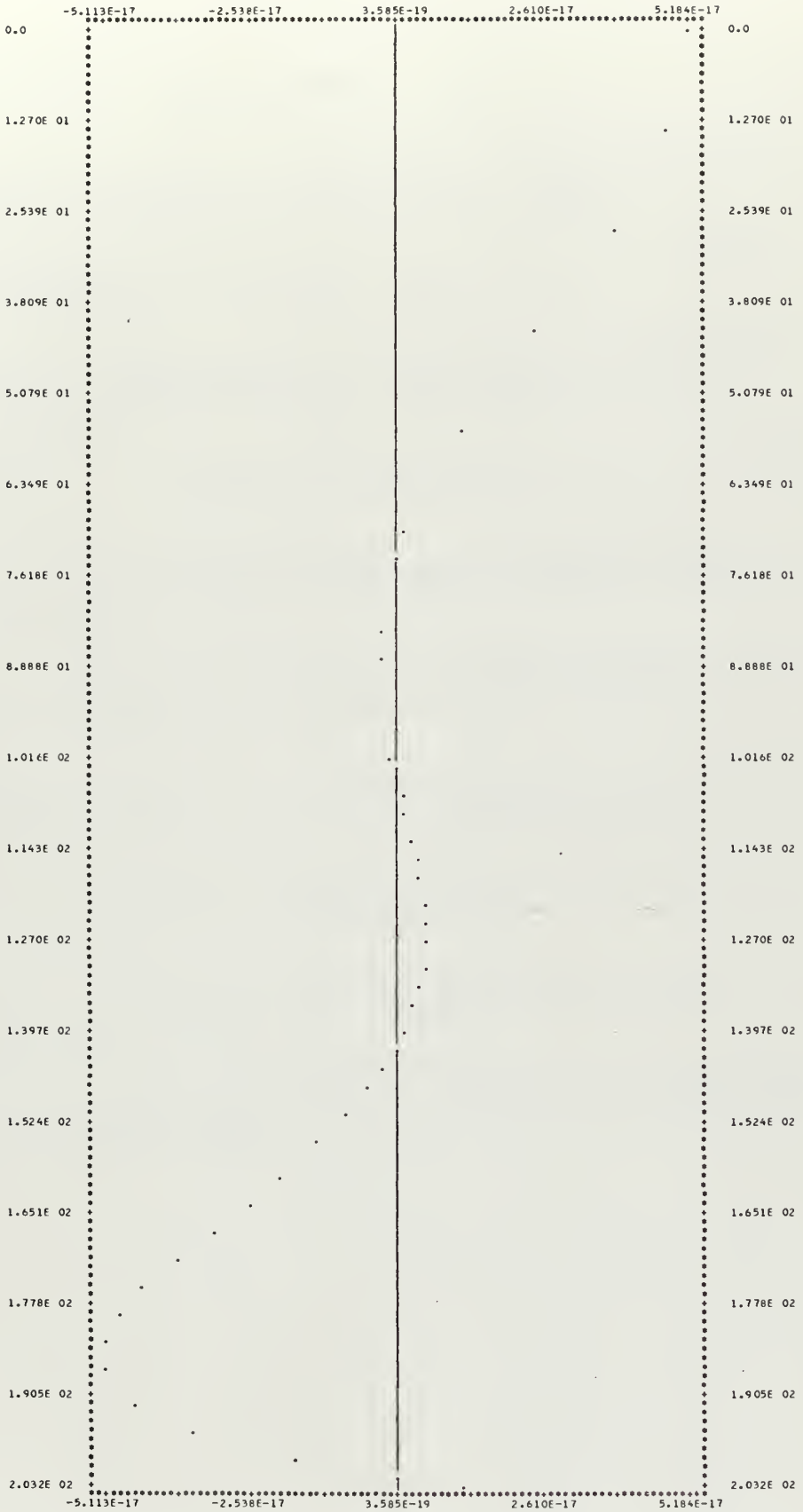
LEFT B.C. CODE= 1
RIGHT B.C. CODE=1
BRANCH # 1 B.C. CODE= 1

****

PIPING SYSTEM NATURAL FREQUENCIES:

MODE	FREQUENCY(RAD/SEC)	FREQUENCY DETERMINANT
1	73.0350449	0.122780-22
2	102.7404907	-.369030-21
3	142.5065200	0.859130-21
4	201.7233189	0.524240-22

GRAPH OF THE VALUE OF FREQUENCY DETERMINANT VS. FREQUENCY



BIBILOGRAPHY

1. Bishop, R. E. D. and D. C. Johnson, The Mechanics of Vibration, Cmabridge University Press, 1960.

2. Brock, J. E., "Compliances of Pipe Components," Naval Engineers Journal, p. 535-538, August 1964.

3. Brock, J. E., "Matrix Analysis of Flexible Filaments," Proceedings of the First U. S. National Congress of Applied Mechanics, p. 285-289, June 1951.

4. Den Hartog, J. P., Mechanical Vibrations, 4th ed., McGraw-Hill, 1956.

5. Fink, G. E., Vibration Analysis of Piping Systems, M.S. Thesis, U. S. Naval Postgraduate School, Monterey, California, 1964.

6. Kim, Y. S., Vibration Analysis of Piping Systems, M. S. Thesis, U. S. Naval Postgraduate School, Monterey, California, 1966.

7. McCracken, D. D. and W. S. Dorn, Numerical Methods and Fortran Programming, John Wiley and Sons, 1964.

8. Ojalvo, I. U. and M. Newman, "Natural Frequencies of Clamped Ring Segments," Machine Design, v. 36, p. 219-222, 21 May 1964.

9. Pestel, E. C. and F. A. Leckie, Matrix Methods in Elastomechanics, McGraw-Hill, 1963.

10. Pestel. E. C. and F. A. Leckie, "Transfer-Matrix Fundamentals," International Journal of Mechanical Sciences, v. 2, p. 137-167, 1960.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor John E. Brock, Code 59 Bc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	2
5.	LT C. D. Rudolf, III 3060 Sunrise Circle Marina, California 93933	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

REPORT TITLE

Vibration Analysis of 3-D Piping via Transfer Matrices

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1971

5. AUTHOR(S) (First name, middle initial, last name)

Chester Davis Rudolf, III

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	139	10

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

This thesis develops the theory for and presents a digital computer program capable of determining the natural frequencies of a three-dimensional piping system having arbitrary configuration. The analysis uses the method of transfer matrices. Piping hangers, loops, and complex branches (branches emanating from branches) have not been included in the analysis. A distributed mass model is used for straight pipe sections, while mass is lumped for curved sections. Inclusion of shear deflection and rotary inertia is optional.

Several piping configurations are analyzed using the program; the results are compared with analytical solutions or values from the literature to demonstrate the accuracy and integrity of the program.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED
Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Transfer Matrix						
Piping						
Vibrations						

Thesis
R835
c.1
Rudolf
Vibration analysis
of 3-D piping via
transfer matrices.
133698